

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

**Étude d'un interface entre un software d'estimation et de simulation écrit en APL  
"IESEM" et un software de simulation écrit en PL1 "SIMUL"  
application dans le cadre du bureau du plan**

Pomes, Yvan

*Award date:*  
1978

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX  
INSTITUT D'INFORMATIQUE  
NAMUR - ANNEE ACADEMIQUE 1977-1978

**ETUDE D'UN INTERFACE**

ENTRE

**UN SOFTWARE D'ESTIMATION ET DE  
SIMULATION ECRIT EN APL " IESEM "**

ET

**UN SOFTWARE DE SIMULATION  
ECRIT EN PL1 " SIMUL ".**

APPLICATION DANS LE CADRE DU BUREAU DU PLAN

**Y. POMES**

MEMOIRE PRESENTE EN VUE DE L'OBTENTION  
DU GRADE DE  
MAITRE EN INFORMATIQUE

FACULTES  
UNIVERSITAIRES  
N.-D. DE LA PAIX  
NAMUR

Bibliothèque

FM B 16  
1978/10

FM B 16 / 1978 110



FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX  
INSTITUT D'INFORMATIQUE  
NAMUR - ANNEE ACADEMIQUE 1977-1978

## **ETUDE D'UN INTERFACE**

ENTRE

**UN SOFTWARE D'ESTIMATION ET DE  
SIMULATION ECRIT EN APL " IESEM "**

ET

**UN SOFTWARE DE SIMULATION  
ECRIT EN PL1 " SIMUL ".**

APPLICATION DANS LE CADRE DU BUREAU DU PLAN

**Y. POMES**

MEMOIRE PRESENTE EN VUE DE L'OBTENTION  
DU GRADE DE  
MAITRE EN INFORMATIQUE



6520-27027



lbs 2013374

Nous tenons à remercier Monsieur H. BOGAERT pour avoir bien voulu accepter et assumer la direction de ce mémoire. La confiance qu'il nous a témoignée, la liberté qu'il nous a laissée et les nombreux conseils qu'il nous a donnés, nous ont permis de réaliser ce travail.

Notre gratitude s'adresse également à tous les utilisateurs APL du Bureau du Plan pour l'accueil qu'ils nous ont réservé et à Mme Lacroix dont la patience a eu raison de tous les algorithmes dessinés dans ce travail.

## TABLE DES MATIERES

INTRODUCTION	5
PRELIMINAIRES	6
1. Le Bureau du Plan dans la procédure de planification en Belgique	6
2. Le Modèle Econométrique, outil technique de la planification	8
3. Quelques modèles utilisés au Bureau du Plan	11
CHAPITRE 1 - LE CADRE INFORMATIQUE DU BUREAU DU PLAN	12
Section 1. L'Organisation du traitement de l'information	12
1. Les objectifs poursuivis	12
2. Le système informatique mis en place	13
Section 2. Le Software IESEM dans ses grandes lignes	16
1. Les concepts de base de IESEM	16
2. Les principales conventions de IESEM	17
3. La routine d'estimation NLEST	19
4. La routine de simulation SIMULATE	22
Section 3. L'application SIMUL, son objectif	23
CHAPITRE 2 - UN TRADUCTEUR D'EQUATIONS ECONOMETRIQUES COMME INTERFACE ENTRE UN SOFTWARE D'ESTIMATION ET DE SIMULATION ECRIT EN APL ET UN SOFTWARE DE SI- MULATION ECRIT EN PL1.	26
CHAPITRE 3 - LEC : UN LANGAGE ECONOMETRIQUE CONDENSE ET SA FORME JUMELLE DBE	32



## INTRODUCTION

Le recours à des modèles empiriques pour la prévision ou pour la simulation de politiques économiques s'est considérablement accru dans tous les pays au cours des dernières années.

Parallèlement on a vu se développer des méthodes d'estimation et d'analyse ayant pour objet de fournir une "évaluation" de ces modèles.

Intégrer l'usage de ces méthodes à l'élaboration même des modèles étudiés permet de conserver une parfaite maîtrise de l'outil mis au point, aux différents stades de son élaboration. Ceci ne va pas sans un effort d'automatisation accrue et la mise au point de programmes standards facilement adaptables.

En Belgique, le Bureau du Plan a été créé pour répondre aux demandes de prévisions du Gouvernement. La réalisation de cet objectif nécessite la conception d'un système informatique capable d'intégrer efficacement les étapes de création et d'utilisation des modèles économiques.

Le système mis en place au Bureau du Plan est double : il comporte d'une part, une partie interactive APL et d'autre part, une partie non-interactive de gestion de base de données IMS (application SIMUL). Ce mémoire se propose de les relier par un software de traduction d'équations économétriques d'un langage de programmation dans un autre. Pour profiter de l'interactivité du langage APL, nous avons écrit ce software également en APL. Une connaissance de base en APL est donc souhaitable pour comprendre certaines références à ce langage.

Notre exposé se divise en quatre chapitres. Au préalable, des préliminaires définissent les contextes économiques et économétriques dans lesquels nous allons nous mouvoir tout au long du travail.

CHAPITRE 4 - LE SOFTWARE DE TRADUCTION	38
Section 1. Son organisation générale	38
Section 2. Les principes de base du traducteur d'une équation économétrique LEC, dans le langage de programmation APL	40
Section 3. Le traducteur d'une équation économétrique LEC, dans le langage de programmation PL1	68
CONCLUSIONS	73
BIBLIOGRAPHIE - REFERENCES	75
ANNEXE 1 - KUKLOS	
Un	
DOSSIER UTILISATEUR DU SOFTWARE	DU.
1. Structure du Software	DU. 1
2. TAPLGP et TPL1GP	DU. 3
3. TRADGP	DU. 7
4. Quelques caractéristiques	DU.11
5. Introduction de nouvelles fonctions ou combinaisons de fonctions	DU.12
et un	
DOSSIER DESCRIPTIF DES FONCTIONS DU SOFTWARE DD1	DD.
1. TAPLGP	DD. 1
2. TPL1GP	DD.75
3. TRADGP	DD.76
sont présentés à la fin de ce mémoire.	

Le chapitre 1 présente le système informatique du Bureau du Plan et constitue le cadre général de référence du présent mémoire.

Le chapitre 2 justifie l'interface que nous avons créé entre les deux systèmes APL et SIMUL.

Les chapitres 3 et 4 décrivent le software de traduction dans ses grandes lignes. Un langage économétrique condensé est la clé de voûte de l'interface, il est défini au chapitre 3. L'algorithme de traduction fait l'objet du chapitre 4.

## PRELIMINAIRES

### 1. Le Bureau du Plan dans la procédure de planification en Belgique.

La Loi du 15 juillet 1970 (Loi Terwagne) "loi-cadre portant organisation de la planification et de la décentralisation économique" a créé le Bureau du Plan et lui a confié la planification de l'économie nationale (1).

Le processus théorique d'élaboration d'un Plan se décompose selon la loi en 2 phases :

- la phase initiale, ou phase des options, permet aux décideurs de fixer les orientations fondamentales du futur Plan, à partir d'un inventaire et d'une confrontation des choix nationaux dits "techniquement possibles" et des options régionales;
- les options retenues doivent ensuite être transposées, lors de la phase de confection du Plan, en un ensemble cohérent et plus détaillé d'objectifs et de moyens.

L'originalité - voulue par le législateur - de la procédure de planification en deux stades et les exigences de concertation qu'il a voulu imposer sont illustrées par le schéma ci-après :

---

(1) Courrier hebdomadaire du C.R.I.S.P. C.H. n° 520-521 - Le 7 mai 1971.



## PROCEDURE D'ELABORATION DU PLAN

PREMIERE ETAPE : LES CHOIX DU PLAN

### BUREAU DU PLAN

LA DIRECTION GENERALE	LA DIRECTION SECTORIELLE	• LES 3 SECTIONS REGIONALES (en collaboration avec les C.E.R. et sur base de l'in- ventaire dressé par les S.D.R.), préparent • LES 3 C.E.R. <u>adoptent</u>
propose	propose	
LES CHOIX DE NIVEAU GLOBAL	LES CHOIX DE NIVEAU SECTORIEL	LES CHOIX DE NIVEAU REGIONAL

confrontation  
AVANT-PROJET D'OPTIONS



### GOUVERNEMENT

Recueille les AVIS du Conseil Central de l'Economie, du Conseil Na-  
tional du Travail et des Conseils économiques régionaux  
les CONCLUSIONS du Comité National de l'Expansion Eco-  
nomique

Délibère, compte tenu de ces avis et conclusions et

Adopte un PROJET D'OPTIONS

Le transmet au Parlement avec l'avant-projet du Bureau du Plan et  
les avis et conclusions prévus ci-dessus



### PARLEMENT

Discute, amende et adopte les OPTIONS du PLAN

DEUXIEME ETAPE : LE PLAN LUI-MÊME

### BUREAU DU PLAN

LA DIRECTION GENERALE	LA DIRECTION SECTORIELLE	• LES 3 SECTIONS REGIONALES (en collaboration avec les C.E.R.), préparent • LES 3 C.E.R. <u>adoptent</u>
prépare	prépare	
LE PROJET DE PLAN AU NIVEAU GLOBAL	LES PROJETS DE PROGRAMMES SECTORIELS	LES PROJETS DE PLANS REGIONAUX

confrontation et intégration  
AVANT-PROJET DE PLAN ARBITRÉ



### GOUVERNEMENT

Recueille les AVIS - des Conseils régionaux  
- des Conseils professionnels  
les CONCLUSIONS du Comité National de l'Expansion Eco-  
nomique

Délibère, compte tenu de ces avis et conclusions et

Adopte le PROJET DE PLAN

Le transmet au Parlement avec les avis et conclusions



### PARLEMENT

Discute, amende et adopte le PLAN

## 2. Le Modèle économétrique, outil technique de la planification

Le modèle économétrique est l'outil de base pour les techniciens de la planification. Quelle est sa nature ? (1)

L'idée de base d'un planificateur est d'imaginer des relations entre les variables de l'économie; prévoyant les unes, il peut tirer des conclusions sur l'évolution des autres.

Ayant défini plusieurs relations, il imagine alors de les grouper pour en former un modèle explicatif de l'économie. La consommation est par exemple supposée égale à un niveau minimum de consommation plus une partie proportionnelle au revenu disponible (revenu - taxe); d'un autre côté, le revenu doit lui-même être égal à ce que l'on consomme plus ce que l'on investit. On a ainsi défini le modèle 1 suivant :

$$\text{Modèle 1 : } \begin{cases} C = \alpha_0 + \alpha_1 (Y - T) \\ Y = C + I \end{cases}$$

où C représente la consommation

Y représente le revenu

T représente les taxes

I représente les investissements

et où  $\alpha_0$ ,  $\alpha_1$  sont les paramètres du modèle à estimer.

Dans un modèle, les variables représentent une série chronologique d'observations (ou de prévisions), soit par exemple la variable

Produit National Brut au prix du Marché, en milliards de F.B.

1960 :	794.859
1961 :	834.260
1962 :	881.541
1963 :	918.616

---

(1) J. Johnston : *Econometric Methods* (2me Edition, 1972, McGraw-Hill).



1964 :	981.915
1965 :	1.020.024
1966 :	1.050.770
1967 :	1.092.366
1968 :	1.139.260
1969 :	1.213.281
1970 :	1.291.824
1971 :	1.346.339
1972 :	1.425.786
1973 :	1.518.022
1974 :	1.592.561
1975 :	1.560.653

Les variables entrant dans un modèle sont classées en :

variables endogènes : ce sont des variables dont la valeur est définie par le modèle (C, Y dans le modèle 1);

variables exogènes : ce sont des variables dont la valeur est définie une fois pour toute à l'extérieur du modèle (T, I dans le modèle 1);

variables prédéterminées : ce sont des variables endogènes, décalées dans le temps; en effet une variable d'une période peut être influencée par sa valeur ou par celle d'une autre variable d'une période précédente.

Un modèle peut enfin se présenter sous plusieurs formes. Dans le modèle 2 suivant :

$$\text{Modèle 2 : } \begin{cases} C_t = \alpha_0 + \alpha_1 (Y_t - T_t) \\ I_t = \beta_1 T_{t-1} + \beta_2 R_t \\ Y_t = C_t + I_t + G_t \end{cases}$$

où  $C_t$  représente la consommation à la période  $t$

$Y_t$  représente le revenu à la période  $t$

$Y_{t-1}$  représente le revenu à la période précédente



$T_t$  représente les taxes à la période  $t$

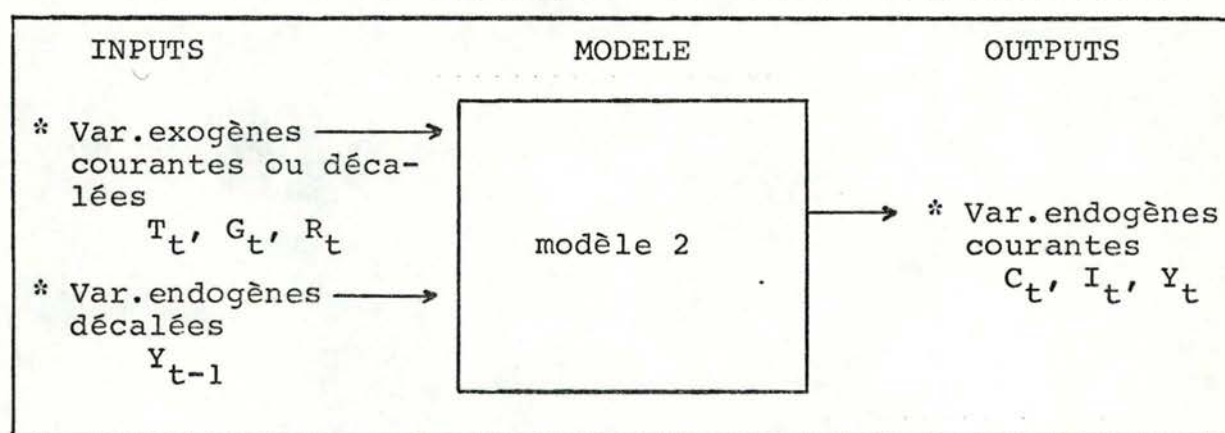
$G_t$  représente les dépenses du Gouvernement à la période  $t$

$I_t$  représente les investissements à la période  $t$

$R_t$  représente le régulateur du Gouvernement à la période  $t$

et où  $\alpha_0, \alpha_1, \beta_1, \beta_2$  sont les paramètres du modèle, à estimer, les variables peuvent être classées comme suit :

Schéma 2

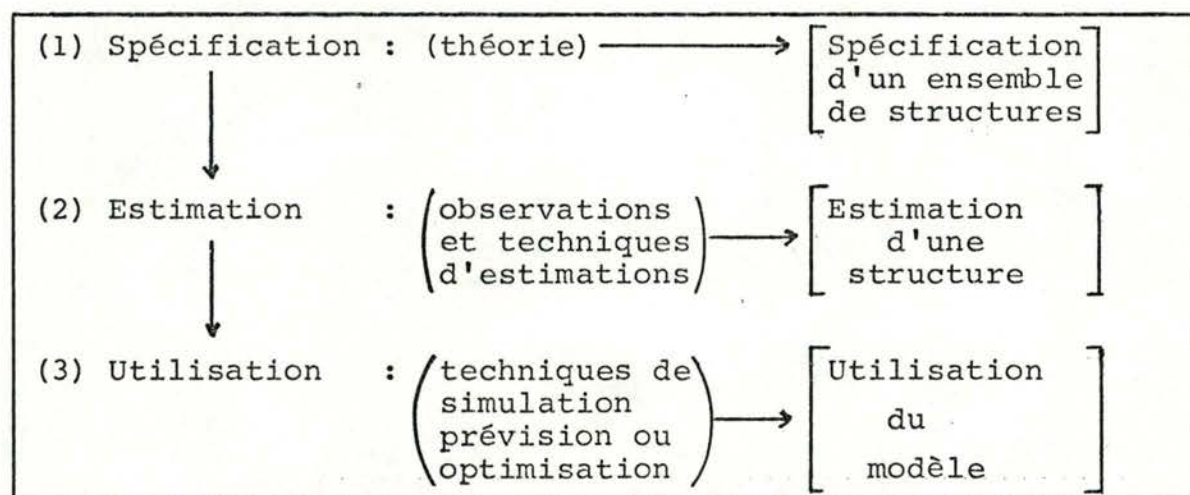


Ce modèle 2 peut être exprimé uniquement en fonction des variables exogènes ou prédéterminées :

$$\text{Modèle 3 : } \begin{cases} C_t = \frac{\alpha_0}{1-\alpha_1} + \frac{\alpha_1 \beta_1}{1-\alpha_1} Y_{t-1} + \frac{\alpha_1 \beta_2}{1-\alpha_1} R_t + \frac{\alpha_1}{1-\alpha_1} G_t - \frac{\alpha_1}{1-\alpha_1} T_t \\ I_t = \beta_1 Y_{t-1} + \beta_2 R_t \\ Y_t = \frac{\alpha_0}{1-\alpha_1} + \frac{\beta_1}{1-\alpha_1} Y_{t-1} + \frac{\beta_2}{1-\alpha_1} R_t + \frac{1}{1-\alpha_1} G_t - \frac{\alpha_1}{1-\alpha_1} T_t \end{cases}$$

Dans ce cas, les coefficients du modèle expriment l'impact de ces variables sur les variables endogènes. Cette formulation est appelée forme réduite du modèle par opposition avec la forme primitive appelée forme structurelle.

De manière générale le processus de construction d'un modèle peut se représenter suivant le schéma ci-dessous :



Ayant rassemblé les données appropriées de l'économie entrant dans le modèle spécifié, l'économetre doit estimer les paramètres du modèle et tester par simulation le modèle estimé : constitue-t-il une image réaliste de l'économie ?

### 3. Quelques modèles utilisés au Bureau du Plan.

Les principaux modèles de planification utilisés par le Bureau du Plan sont :

- les modèles nationaux à répartition régionale, c'est le cas de RENA (Regional-National), l'actuel modèle du Bureau du Plan;
- les modèles régionaux à cohérence sectorielle et nationale. C'est le cas de SERENA.

D'autres grands modèles sont également mis en chantier, tels sont Planet, Comet, ...

Enfin, de nouvelles relations (fonctions de production, ...) ou des modèles de taille plus réduite, destinés à la prévision, sont continuellement mis au point au Bureau du Plan. Tel est



par exemple le modèle KUKLOS; c'est sur lui que nous avons testé le software de traduction (1).

## Chapitre 1 - LE CADRE INFORMATIQUE DU BUREAU DU PLAN

### Section 1 - L'organisation du traitement de l'information

#### 1. Les objectifs poursuivis

Pour répondre aux demandes de prévisions du Gouvernement, le Bureau du Plan met continuellement à jour les modèles existants ou en crée de nouveaux. Ce travail d'élaboration d'un modèle peut comprendre une ou plusieurs des étapes suivantes :

STEP0 : Conception du modèle

- quelles variables ?
- comment les relier ?

STEP1 : - Construction et recherche des données  
- Spécification des équations

STEP2 : - Estimation des coefficients

STEP3 : - Structuration en modèle Simulable  
- Simulations statiques et dynamiques sur le passé  
- Prévisions des exogènes + coefficients  
- Etablissement d'une projection de référence  
- Etude de projections alternatives  
- Etude de scénarios

(1) Une note du Bureau du Plan, présentant le modèle KUKLOS (30.9.1977), est donnée en Annexe 1.



Pour cela, le Bureau du Plan a développé un système informatique ayant pour objectifs :

1) d'intégrer les étapes de modélisation :

les activités de création, d'estimation, de test et d'utilisation des modèles doivent être effectuées en concordance; l'interface d'une étape à la suivante doit être complètement automatique.

2) de rendre efficaces les conditions de travail :

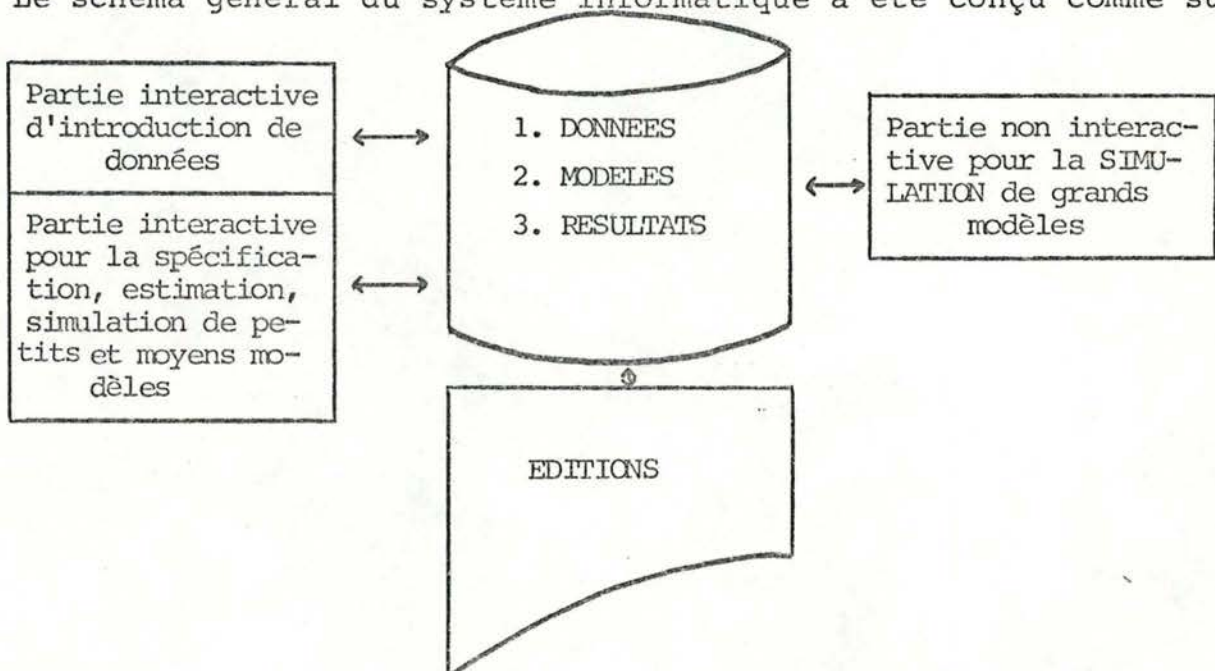
cela implique d'une part un software de construction de modèles, interactif et couvrant l'ensemble des étapes de modélisation; d'autre part il faut que le software soit accessible à plusieurs utilisateurs, en même temps.

Ceux-ci doivent avoir :

- un bon temps de réponse lorsqu'ils questionnent un modèle;
- une utilisation simultanée d'un modèle;
- une possibilité d'outputs variés;
- une personnalisation de l'utilisation et du contrôle du système.

2. Le système informatique mis en place.

Le schéma général du système informatique a été conçu comme suit :



Un système APL supporte toute la partie interactive, un système IMS-D.B.M. (application SIMUL) supporte toute la partie base de données à laquelle s'alimente directement la partie non-interactive et indirectement la partie interactive.

Actuellement, il n'existe pas d'interface entre ces deux systèmes APL et IMS, ce qui oblige le système APL à disposer de ses propres fichiers de données. Celui-ci se décompose donc en un software de Calcul (IESEM = Interactive Estimation and Simulation of Econometric Models (1)) et un software de gestion des fichiers (TSAR = Time Séries APL Retrieval (2)).

La partie IMS est quant à elle composée de 6 bases de données sur lesquelles se greffent un software de Calcul non-interactif (SIMUL (3)).

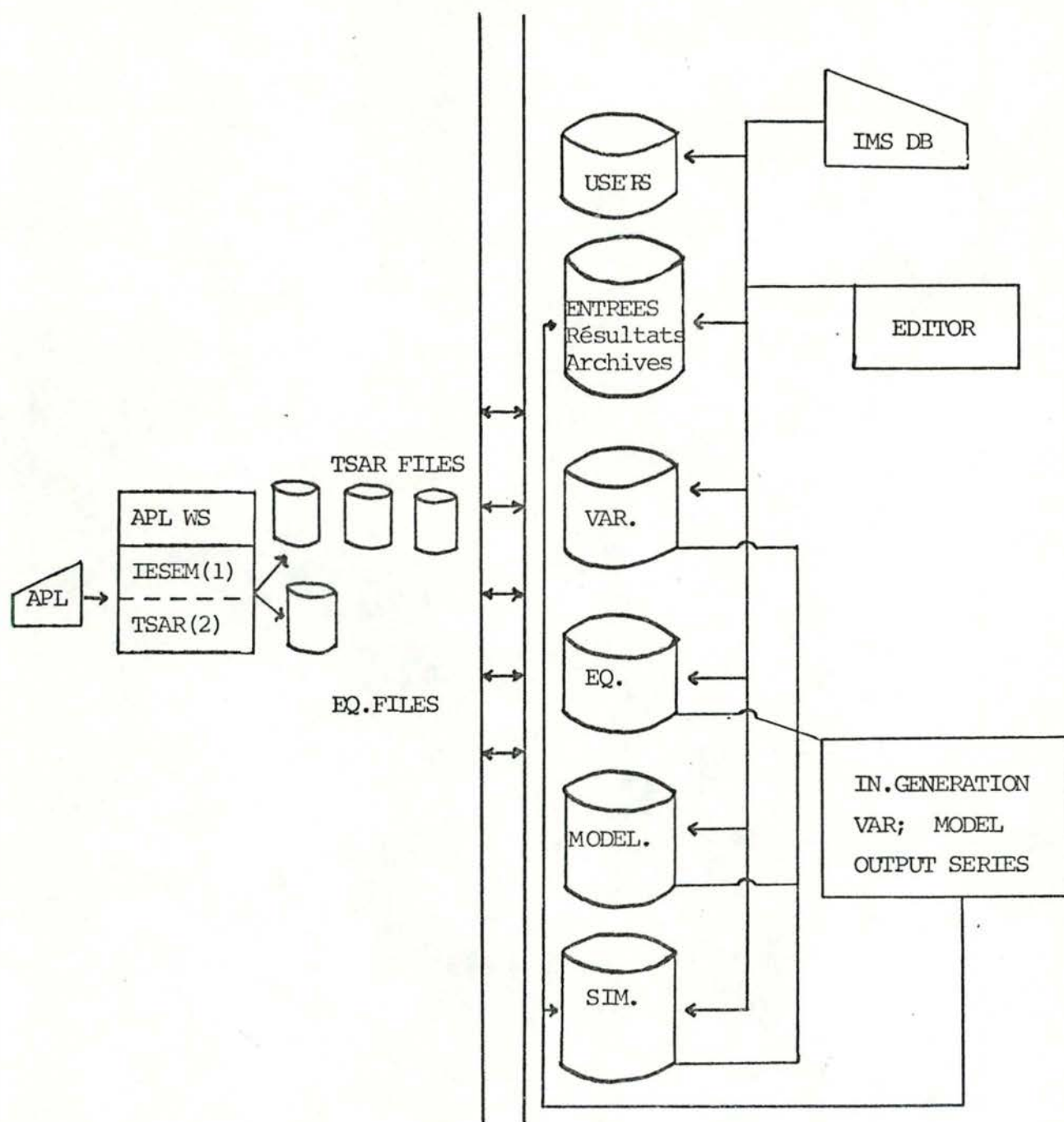
---

(1) Dossier IESEM : Bureau du Plan. Référence (77)HS-1491/ev/1617.

(2) Dossier TSAR : Bureau du Plan. Référence (77)JW-AB/1373/hg/1488.

(3) Dossier SIMUL : Bureau du Plan. Référence (76)DK-HJB/897/ev/942.

Schématiquement ces 2 systèmes parallèles se présentent ainsi :



BD1 : users (utilisateurs du système)

BD2 : variables : - séries observées

- séries de référence (décidées)

- séries prévisionnelles (utilisateurs)

+ commentaires

BD3 : équations : - chaque équation pointe vers les variables explicatives nécessaires à son estimation ou à sa simulation

BD4 : modèles : un article modèle pointe vers les équations qui le composent ou vers un autre modèle.



Il existe donc un double problème d'interface entre ce qui se fait de manière interactive et les bases de données IMS.

- Il faudrait d'une part que le système APL puisse s'alimenter directement à la base de donnée VAR de IMS, sans passer par la création "manuelle" de fichiers de variables à structure TSAR.
- Il faudrait d'autre part que les équations économétriques mises au point interactivement en APL puissent être rendues directement utilisables par SIMUL.

Notre objectif d'interface ne vise que ce second problème.

## Section 2 - Le software IESEM, dans ses grandes lignes

La partie APL du support informatique du Bureau du Plan est constituée :

- d'un système de fichiers de données : les fichiers TSAR;
- d'un système exploitant les données fournies par le système TSAR : le Software IESEM.

Ce software constitue le cadre dans lequel s'inscrit la suite du travail. La première section définit les concepts de base, la seconde décrit les principales conventions de langage utilisées. Les sections 3 et 4 présentent les 2 fonctions principales de ce software : NLEST, la fonction d'estimation des équations économétriques et SIMULATE, la fonction de simulation de ces équations.

### 1. Les concepts de base de IESEM

IESEM travaille à partir :

- d'un "fichier" d'équations : ce "fichier" est un tableau alphanumérique comportant autant de lignes qu'il y a d'équations;

- d'un "fichier" de données : ce "fichier" est un tableau alphanumérique contenant les valeurs des différentes variables; chacune occupe une ligne de cette matrice;
- d'un "fichier" de construction : ce "fichier" est un tableau alphanumérique contenant l'information nécessaire pour accéder au système TSAR, et construire le fichier de données. A chaque ligne correspond une équation définissant le nom d'une variable par les symboles utilisés dans TSAR.

L'utilisation de tels "fichiers" alphanumériques peut paraître un détour inutile; elle présente l'inconvénient d'exiger un espace mémoire accru, cependant elle facilite la manipulation et la présentation de groupes de variables et d'équations; d'autre part, elle rend les opérations plus sûres : au sein d'un "fichier", les vecteurs numériques sont à l'abri d'une transformation involontaire; quant aux "fichiers" d'équations, la procédure correspond à une rationalisation appréciable du fonctionnement de nombreuses fonctions du software IESEM (1).

## 2. Les principales conventions de IESEM

En ce qui concerne les variables, l'unique contrainte est la présence d'un Header SIMUL, comme première observation de la série. Le Header SIMUL est un ensemble de chiffres caractérisant la série; il est composé d'un indicatif de périodicité, de l'année de départ de la série et de la période de départ dans l'année pour les séries de périodicité infra-annuelle :

ex.: une variable de Header 01|1954|01 représente donc une série d'observations annuelles commençant en 1954.

---

(1) Dossier IESEM : Bureau du Plan. Référence (77)HS-1491/ev/1617.



En ce qui concerne les équations, elles doivent être écrites comme une quelconque instruction APL d'affectation : cela implique notamment qu'aucun opérateur ne peut être appliqué à la variable dépendante de l'équation. C'est le littéral de ces équations d'affectation qui est stocké dans les "fichiers" d'équation.

De plus les conventions de langage suivantes sont imposées :

- les noms de variables doivent être constitués uniquement de lettres non soulignées ou de chiffres, le premier caractère devant être une lettre.

ex.: CP0 = consommation privée à prix constants;

- les lettres soulignées sont obligatoirement réservées aux noms de fonctions APL;
- chaque nom de variable doit être immédiatement suivi de l'indice temps [I]. Les retards doivent être indiqués et précédés du signe - de soustraction.

ex.: CP0[I] = consommation publique à prix constants à l'année I

CP0[I-1] = consommation publique à prix constants à l'année I - 1

Cet indice de temp[I-n] doit être supérieur ou égal à 1, ce qui n'est pas toujours le cas en APL où l'on peut définir un vecteur de 0 à n-1 pour représenter n valeurs (  $\square$  IO = index origine; il peut valoir 0 ou 1 ).

L'utilisation obligatoire de l'indice temps a des désavantages évidents : elle nécessite l'écriture détaillée de chaque expression, vu qu'il n'existe pas de fonctions calculant automatiquement des différences premières ou des taux de croissance mais que cela se fait par un indice qui doit suivre immédiatement chaque nom de variable. Cette obligation a cependant l'avantage de ne prendre en considération que la partie utile d'un vecteur. Elle permet également de manipuler des vecteurs de longueurs différentes sans devoir calculer à chaque opération leur intersection.



Une exception à cette obligation concerne le "fichier" de construction qui doit, lui, être défini en fonction du système TSAR.

- Les coefficients à estimer doivent être renseignés comme constituant les éléments d'un vecteur C. A chaque équation (ou groupe d'équations, si l'on demande une estimation jointe de ces équations (1)), doit correspondre un vecteur différent ex.: C[1], C[2], C[3].

### 3. La routine d'estimation NLEST

NLEST est une routine d'estimation d'équation non linéaire (Non Linear ESTimation). La justification théorique de cette méthode d'estimation est donnée dans l'annexe 1 du dossier IESEM (2).

La méthode d'estimation développée tire parti de la manière dont sont présentées les équations. Les coefficients à estimer étant toujours symbolisés par les éléments d'un vecteur C, il est aisé d'approximer numériquement les dérivées des équations par rapport à chaque coefficient. Dans le cas linéaire, l'approximation est exacte.

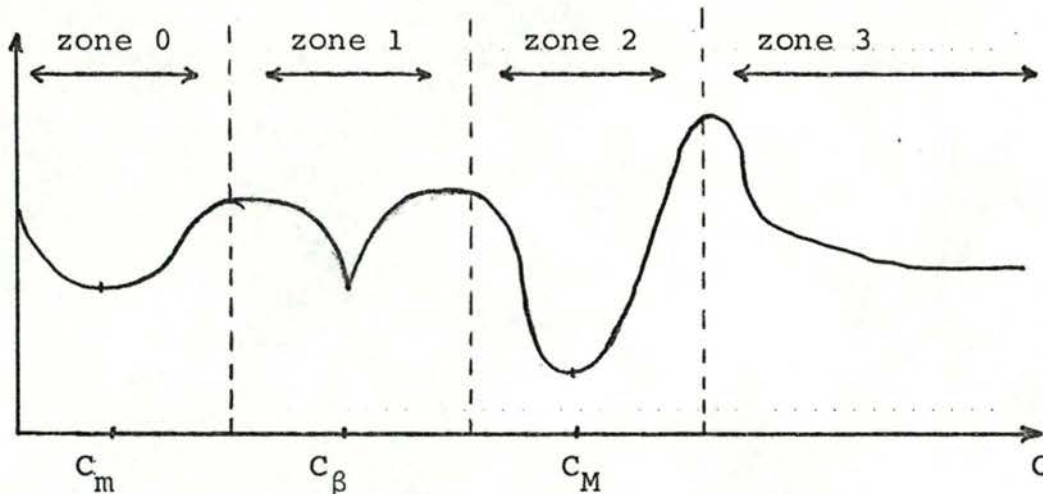
Dans le cas non linéaire, le processus d'estimation est itératif; la matrice des dérivées (approximées) n'est plus équivalente à la matrice des observations, mais permet le calcul d'une correction à effectuer sur les valeurs initiales attribuées aux coefficients  $C = C + SMO \times \Delta C$ ; la même procédure est répétée jusqu'à convergence, c'est-à-dire jusqu'au moment où la correction à effectuer est devenue suffisamment petite. A ce point, les estimateurs obtenus minimisent une somme pondérée des résidus des équations.

---

(1) Cf. La routine d'estimation NLEST.

(2) Dossier IESEM. Bureau du Plan. Référence (77)HS-1491/ev/1617.

Cette présentation du processus itératif invite à souligner quelques cas susceptibles de se produire vu la forme aléatoire de la fonction de correction des coefficients : ce sont :

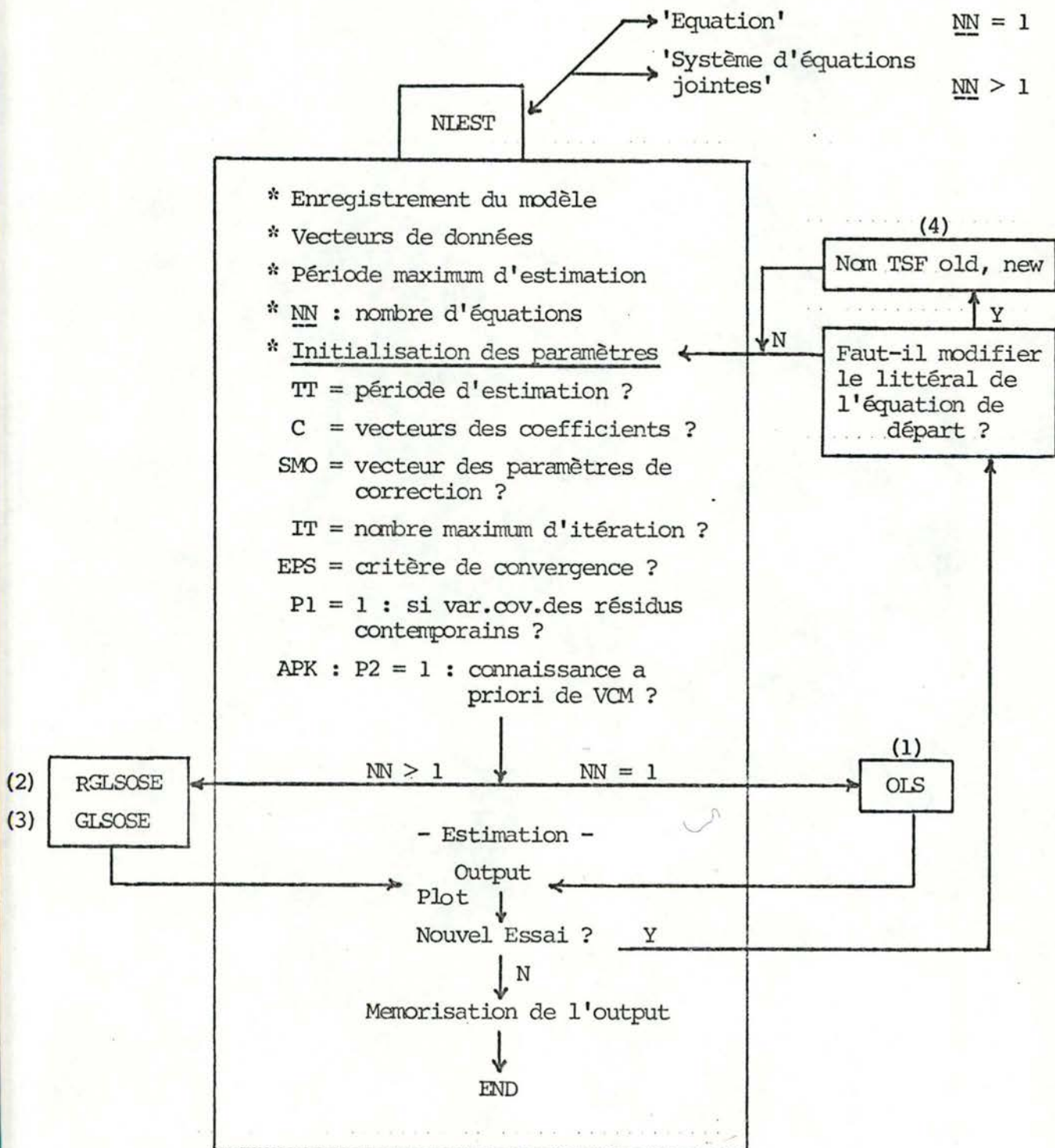


- la présence de plusieurs minima : rien ne permet alors d'assurer que le minimum atteint soit un minimum global =  $C_m$
- l'absence de minimum et non convergence : c'est le cas de la zone 3
- une fonction non-dérivable en un point :  $C_\beta$  : on repasse continuellement par les mêmes valeurs des coefficients sans parvenir à stabiliser le processus. Pour résoudre ce problème, il faut fixer le coefficient à une certaine valeur de départ et l'empêcher de changer au cours des itérations.

Outre des équations comportant des coefficients connus et/ou fixés a priori, outre des équations linéaires et/ou non linéaires dans les paramètres et/ou les variables, la routine NLEST veut résoudre les systèmes d'équations à estimer de façon jointe : dans ce cas, d'une part, le vecteur des coefficients doit être défini de manière unique pour tout le système, mais d'autre part, la routine doit tenir compte d'une corrélation éventuelle des résidus contemporains.



Le schéma général d'organisation de NLEST répond à l'objectif d'interactivité et de souplesse dans la construction des modèles économétriques :



(1) OLS = Ordinary Least Squares

(2) RGLSOSE = Restricted Generalized Least Squares On a Set of Equations (résidus indépendants)

(3) GLSOSE = Generalized Least Squares On a Set of Equations

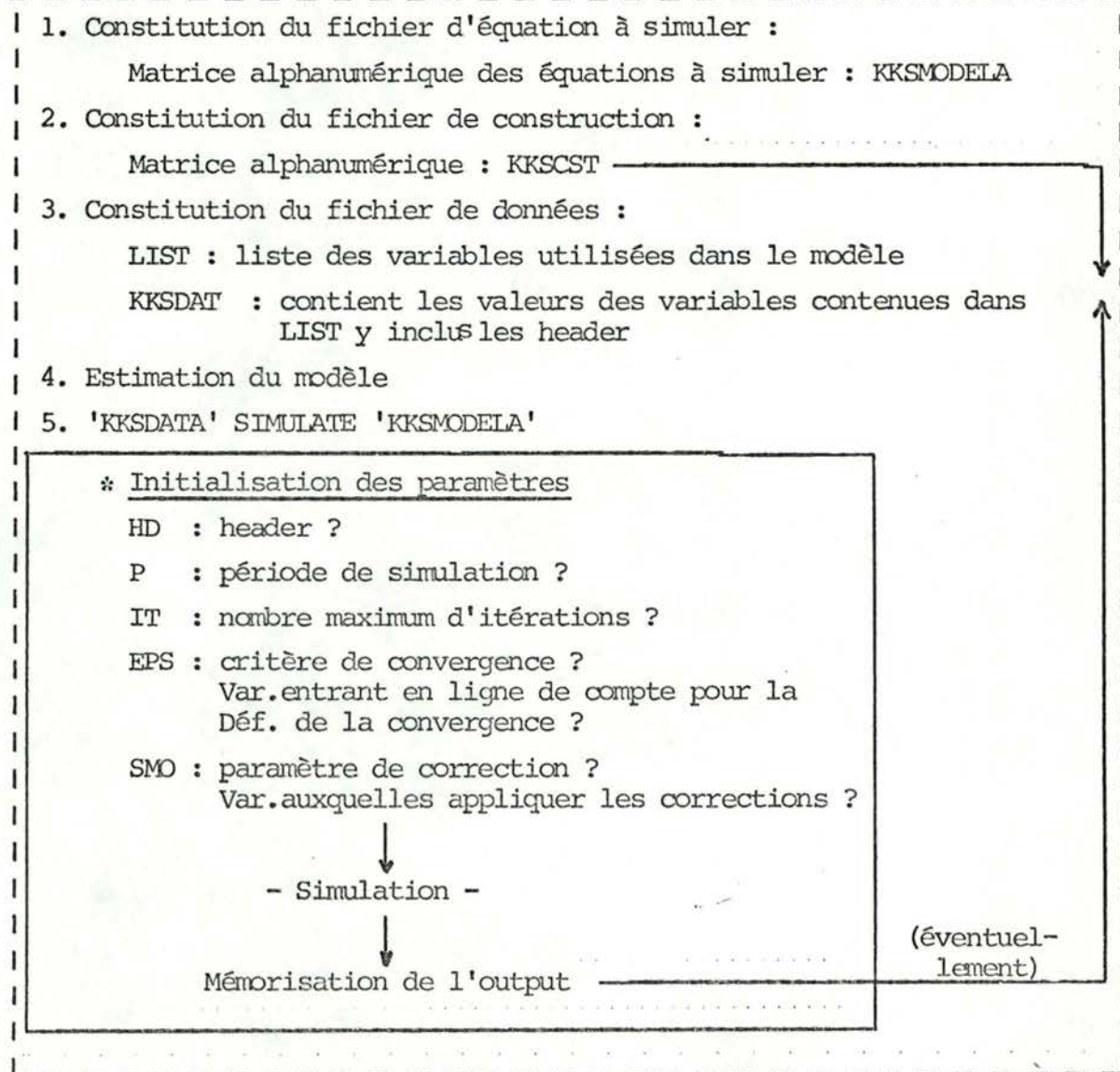
(4) TSF : transforme le lit.d'l équation en un autre littéral.



#### 4. La routine de simulation : SIMULATE

SIMULATE est une routine de simulation réalisée par un processus itératif suivant la méthode de Gauss-Seidel. La convergence est atteinte lorsque d'une itération à l'autre, les valeurs des variables dépendantes (ou d'une partie d'entre elles) restent suffisamment stables.

Le schéma général d'organisation de simulate est le suivant :



### Section 3 - L'application simul, son objectif

L'objectif du projet simul est l'élaboration d'un système capable de réaliser la constitution, la simulation, l'archivage et la documentation de modèles économétriques en général. Ceci suppose la suppression de la contrainte de liaison étroite d'une équation économétrique à un modèle précis. Il ne peut plus être question d'entrer un modèle sous forme d'une sous-routine contenant les équations dans leur forme explicite; la moindre modification du modèle pour des simulations différentes serait très lourde.

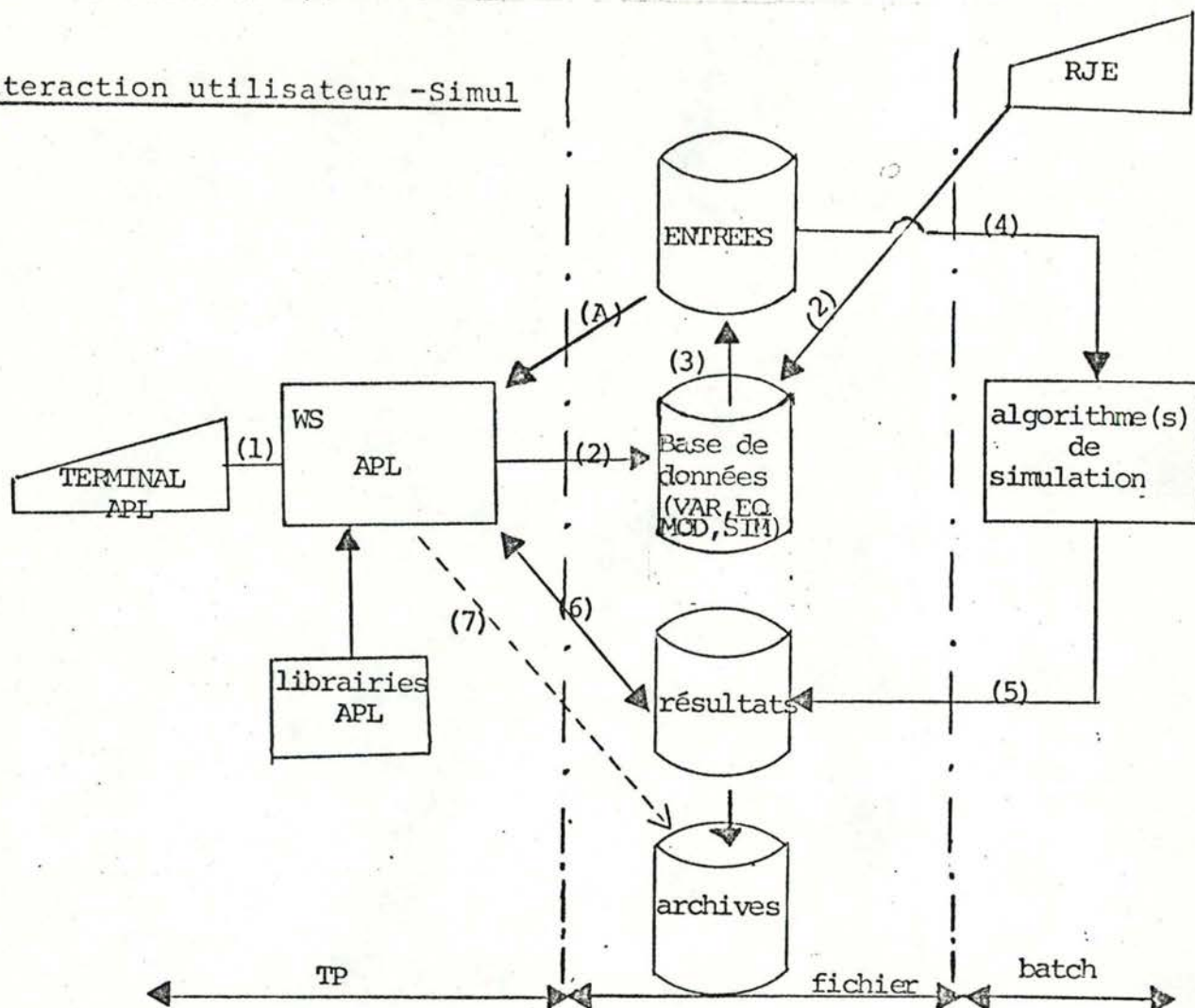
Dans le nouveau système, les variables aussi bien que les équations et les paramètres de simulation constituent des données (entrées) par la routine de simulation; chaque personne peut ainsi s'intéresser à son propre modèle et exécuter les simulations qu'elle désire sur les séries qu'elle veut.

La création de fichiers permanents de variables/d'équations/de modèles/de paramètres de simulation, l'interactivité et la facilité du langage et du système APL, la performance des langages traditionnels pour l'exécution des différents algorithmes de résolution des modèles, devraient servir la réalisation de cet objectif.

L'interaction Utilisateur-SIMUL peut se schématiser comme suit :



# Interaction utilisateur - Simul



- (1) L'utilisateur de son terminal a accès à l'APL, aux différentes librairies du système APL, et aux divers fichiers
- (2) Des fonctions APL lui permettront d'aller alimenter les data bases (notamment le fichier des équations qu'il pourrait estimer avec IESEM et les introduire ensuite dans le fichier). La création, l'alimentation et la mise à jour des fichiers se fera autant par batch que par téléprocessing.
- (3) A partir des fichiers de variables, d'équations, de modèles et de paramètres de simulation, l'utilisateur pourra créer
- (4) son propre fichier d'entrées qui servira d'input au(x) programme(s) de simulation contenant le(s) algorithme(s) de simulation).

Le fichier d'entrées contiendra toutes les informations nécessaires aux simulations

- paramètres de la simulation
- code du modèle



- liste des équations en séquence correspondant à l'ordre de résolution
  - liste des séries chronologiques relatives aux variables renfermées dans les équations.
- (5) Après résolution du modèle, les résultats sont écrits sur un fichier RESULTATS qui pourra être consulté par l'utilisateur et analysé (par exemple par des fonctions APL ou des programmes d'analyse batch).
- (6) L'utilisateur pour l'analyse des résultats de sa simulation pourra amener en work space (dans le cas du travail en APL) les séries résultats qu'il désire prendre en considération.
- (7) S'il estime que les résultats sont bons, il peut les archiver et dans la suite exploiter ce fichier par des programmes d'édition et d'analyse.
- (A) Consultation du fichier d'Entrées par APL. Le fichier d'entrées sera accessible en lecture uniquement pour l'utilisateur.

Les 4 fichiers centraux sont

1. Data base variables qui contralise les données d'entrée de tous les modèles et simulations.  
Des données simulées (provenant du fichier résultat) peuvent être réinjectées dans la data base pour prolonger des séries observées par exemple.
2. Data base équations les équations seront stockées comme des données sous leur spécification SIMUL et/ou IESEM avec documentation obligatoire s'il s'agit d'une équation estimée.
3. Data base modèles : contient la liste des équations et/ou du/ des modèle(s) dont un modèle est composé et l'ordre dans lequel il doit être simulé.
4. Data base Simul : contient les paramètres nécessaires à une simulation et ce pour toutes les simulations.

## Chapitre 2 - UN TRADUCTEUR D'EQUATIONS ECONOMETRIQUES COMME INTERFACE ENTRE UN SOFTWARE D'ESTIMATION ET DE SIMULATION ECRIT EN APL ET UN SOFTWARE DE SIMULATION ECRIT EN PL1

Nous avons décrit au chapitre précédant l'objectif de l'application SIMUL, à savoir l'élaboration d'un système capable de réaliser la constitution, la simulation, l'archivage et la documentation de modèles économétriques en général.

Or ces modèles sont mis en chantier interactivement au terminal APL et l'on voudrait les utiliser aussi bien dans la partie non-interactive écrite en PL1, qu'au terminal APL.

La réalisation de l'objectif d'intégration de l'application SIMUL nécessite non seulement la création de bases de données communes aux deux systèmes APL et SIMUL, mais doit donc encore résoudre le problème du passage d'une équation économétrique d'un langage de programmation vers un autre.

Une équation doit en effet, pouvoir entrer dans un modèle simulable par le Software de calcul de SIMUL écrit en PL1, mais elle doit également rester disponible en APL pour être réestimée, modifiée ou complètement réécrite.

C'est à ce problème que nous allons essayer d'apporter une solution.

Actuellement ce passage d'une équation APL vers un modèle PL1 se fait comme suit :

1. L'équation spécifiée est écrite en APL par l'utilisateur.
2. Son littéral est stocké dans un "fichier" d'équations sur le disque APL.



3. Reprise du disque APL, elle doit être traduite par l'utilisateur en syntaxe PL1.
4. Elle est entrée par terminal IMS DB/DC dans SIMUL.

Voyons ce que chacun de ces points implique :

0. Tout d'abord, cette manière de procéder oblige l'utilisateur à connaître deux syntaxes arithmétiques différentes : APL et PL1.
1. La syntaxe APL (1) présente la difficulté d'une évaluation des expressions basées sur la seule règle d'association à droite. Prenons par exemple, l'évaluation de :
  - .  $12 + 3 - 4 - 5 = 6$  en arithmétique normale
  - $12 + 3 - 4 - 5 = 16$  en APL
  - .  $\text{LOG}100 + 900 = 902$  en arithmétique normale
  - $\text{LOG}100 + 900 = 3$  en APL

En observant les "fichiers" d'équations créés par les utilisateurs, on remarquera qu'ils tournent tout normalement la difficulté d'évaluation de Droite à Gauche par une introduction massive de parenthèses. Or ces équations sont destinées à être évoluées itérativement par NLEST tout d'abord, par SIMULATE ensuite.

Prenons comme exemple, l'équation suivante :

$$(1) \text{LNQAFF}[I] \leftarrow (\text{QKF}[I]) + (C[1] \times \text{TEMPS}[I] \times (((\text{QIG}[I-1] - \text{QIG}[I-3]) \div (\text{QIG}[I-4] - \text{QIG}[I-6])) * .3))$$

réécrite elle devient :

$$(2) \text{LNQAFF}[I] \leftarrow (\text{QKF}[I]) + C[1] \times \text{TEMPS}[I] \times ((\text{QIG}[I-1] - \text{QIG}[I-3]) \div \text{QIG}[I-4] - \text{QIG}[I-6]) * .3$$

Etant donné le nombre de parenthèses présentes, l'évaluation de (1) nécessite 7 appels récursifs d'évaluation d'une expression  
8 opération arithmétiques (sans compter le calcul des indices de temps).

---

(1) APL : W. Giloř. Programmieren in APL.



L'évaluation de (2) qui utilise au mieux la syntaxe APL nécessite : 3 appels récursifs d'évaluation d'une expression

8 opérations arithmétiques (sans compter le calcul des indices de temps).

2. Si l'utilisation de "fichiers" d'équations facilite la présentation et la manipulation de groupes de variables et d'équations, elle exige cependant un espace mémoire accru.
3. Ayant mis au point une équation en APL, l'utilisateur doit la retraduire "à la main" en PL1; il se heurte tout d'abord au "déchiffrage" de cette équation APL avant d'être confronté à une nouvelle syntaxe, celle-ci évaluant les expressions de gauche à droite.

Prenons comme exemple la traduction de :

$$\begin{aligned}
 \text{WIR}[I] \leftarrow & * (\text{QWIR}[I-1]) + C[1] + (C[2] \times (\text{QWIR}[I]) - \text{QWIR}[I-1]) \\
 & + ((1 - C[2]) \times (\text{QWIR}[I-1]) - \text{QWIR}[I-2]) \\
 & + C[3] \times ((\text{YY}[I] - \text{IT}[I] + \text{WE}[I]) \div \text{FY}[I]) \\
 & - (1 - C[2]) \times (\text{YY}[I-1] - \text{IT}[I-1] + \text{WE}[I-1]) \div \text{FY}[I-1]) \\
 & + C[4] \times ((\text{QDUC}[I] - \text{QDUC}[I-1]) - (1 - C[2]) \times ((\text{QDUC}[I-1]) - \text{QDUC}[I-2]))
 \end{aligned}$$

En PL1, elle devient :

#### 4. La nouvelle équation PL1 est entrée par terminal IMS dans SIMUL.

La réalisation de l'intégration devrait donc passer par un traducteur automatique d'une syntaxe dans une autre, et le stockage d'une ou éventuellement plusieurs syntaxes dans la Base de données des équations (DBE).

Deux solutions se présentent à première vue :

##### 1) Un traducteur d'une syntaxe APL vers une syntaxe PL1.

L'inconvénient majeur est que l'utilisateur doit toujours parfaitement dominer la syntaxe Droite-Gauche de APL. De plus le problème de lisibilité d'une équation en syntaxe APL existe toujours.

##### 2) Un traducteur d'une syntaxe PL1 vers une syntaxe APL.

L'utilisateur a l'avantage de travailler dans une syntaxe évaluant ses expressions selon les règles de l'arithmétique normale. Cependant, travaillant à un terminal APL, l'utilisateur est obligé de dominer une syntaxe étrangère à APL.

Plus encore qu'un traducteur d'une syntaxe d'un langage de programmation dans une autre, on pourrait profiter de la création d'un traducteur pour se doter d'une syntaxe spéciale qui correspond directement aux besoins de l'économètre. Le traducteur partirait alors de cette syntaxe pour générer le littéral de l'équation dans un langage de programmation donné. Cette nouvelle syntaxe devrait satisfaire trois objectifs:

- 1) présenter un littéral aisément lisible et compréhensible par un utilisateur quelconque de l'équation;
- 2) être aussi proche que possible d'une notation économétrique courante, être entre autres, d'une syntaxe arithmétique normale et indépendante d'un langage de programmation;
- 3) être enfin facilement manipulable par le concepteur d'équations économétriques; entre autres, le jeu d'opérateurs économétriques propre à cette nouvelle syntaxe doit pouvoir



être étendu, par l'utilisateur lui-même, au fur et à mesure de ses besoins.

C'est cette solution que nous avons adoptée.

Nous avons appelé la nouvelle syntaxe : LEC : Langage Econométrique Condensé; elle répond principalement aux objectifs 2 et 3.

Pour répondre plus précisément à l'objectif 1 de lisibilité, nous avons créé une forme jumelle à LEC, la forme DBE : c'est cette forme jumelle qui sera stockée dans la base de données d'équations. Elle est notamment destinée à être imprimée en Remote Batch : contrairement à LEC, elle ne contient aucun caractère souligné, ceux-ci ne faisant pas partie du jeu de caractères disponibles sur le terminal lourd.

Avant de décrire ces deux syntaxes de manière détaillée au chapitre suivant, voyons quelques exemples illustrant ces deux formes LEC et DBE :

- Soit VAR le nom d'une variable.

Nous avons voulu dispenser le concepteur d'équations économétriques de l'obligation d'écrire l'indice de temps [I] à la suite de chaque nom de variable. VAR est donc équivalent à VAR[I].

- De plus, nous avons voulu être aussi proche que possible d'une notation économétrique courante. C'est ainsi que

pour LEC

pour DBE

VAR(1)

VAR(1)

représente cette variable décalée d'une période dans le temps; elle est donc équivalente à VAR[I-1]



<u>L</u> VAR	LOG : VAR	représente le logarithme de cette variable; elle est équivalente à LOG(VAR[I])
<u>D</u> VAR	DIF : VAR	représente la différence première de cette variable; elle est équivalente à VAR[I] - VAR[I-1]
n <u>D</u> VAR	n DIF : VAR	représente la différence nième de cette variable; elle est équivalente à VAR[I] - VAR[I-n]
2 <u>DL</u> VAR	2 DIF : LOG : VAR	représente la différence deuxième du loga ithme de cette variable; elle est équivalente à LOG(VAR[I]) - LOG(VAR[I-2])
<u>C</u> 1	C : 1	représente le coefficient C[1] de l'équation à estimer.

C, L, D, ... (C:, LOG :, DIF :, ...) font partie du jeu des opérateurs économétriques propres à la syntaxe LEC(DBE); ils peuvent porter sur une variable ou sur une expression.

- Enfin une équation se présente comme suit :

Equation 1

Forme LEC : L VAR = C 1 + C 2 x L VAR1

Forme DBE : LOG : VAR = C:1 + C:2 x LOG:VAR1

Equation 2

Forme LEC : DL WIR = C 1  
+ C 2 x (L WER - L WIR(1))  
+ (1-C2) x DL WIR(1)

$$\begin{aligned}
 \text{Forme DBE : DIF : LAG : WIR} &= C:1 \\
 &+ C:2 \times (\text{LOG:WER} - \text{LOG:WIR}(1)) \\
 &+ (1-C:2) \times \text{DIF:LOG:WIR}(1).
 \end{aligned}$$

### Chapitre 3 - LEC : UN LANGAGE ECONOMETRIQUE CONDENSE ET SA FORME JUMELLE DBE

Pour réaliser l'intégration du système APL dans l'application SIMUL, nous nous sommes tournés vers un programme traducteur d'une syntaxe dans une autre.

Mais plus encore qu'un traducteur d'un langage de programmation dans un autre, nous avons voulu profiter de cette occasion pour définir un langage qui correspondrait directement aux besoins de l'économetre. C'est ce langage LEC, et sa forme jumelle DBE que nous allons décrire maintenant.

Les symboles de base utilisés sont :

pour LEC

ABCDEFGHIJKLMN
OPQRSTUVWXYZ
ABCDEFGHIJKLMN
OPQRSTUVWXYZ (1)
0123456789
+ - x ÷ * ( ) = < ≤ ≥ > ≠ VΛ~.

pour DBE

ABCDEFGHIJKLMN
OPQRSTUVWXYZ
:
0123456789
+ - * ( ) = .

Les différents éléments de ce langage sont :

- les noms de variable
- les numériques
- les opérateurs binaires arithmétiques et booléens
- l'opérateur de retard
- les fonctions ou opérateurs spécifiques au problème économétrique

(1) Le terminal APL offre la possibilité de souligner une lettre de façon à étendre le jeu de caractère disponible. Conformément au Software IESEM, tout nom de fonction sera souligné et les lettres soulignées ne seront utilisées qu'à définir des fonctions.

## 1. Les noms de variables.

Une variable représente une série chronologique d'observations statistiques ou de prévisions.

Le nom d'une variable est composé de lettres ou de chiffres, le premier devant être une lettre.

## 2. Les numériques.

Les numériques sont des nombres composés des chiffres 0123456789 et du point décimal.

## 3. Les opérateurs binaires sont :

- les opérateurs binaires arithmétiques de

	<u>pour LEC</u>	<u>pour DBE</u>	
l'addition	+	+	de priorité 1
la soustraction	-	-	de priorité 1
la multiplication	x	*	de priorité 2
la division	÷	/	de priorité 2
l'exposant	*	**	de priorité 3

En cas d'opérateur de priorité égale, l'évaluation est supposée se faire de gauche à droite.

- les opérateurs

	<u>pour LEC</u>	<u>pour DBE</u>	
plus petit	<	.LT.	de priorité 4
plus petit ou égal	≤	.LE.	de priorité 4
égal	=	.EQ.	de priorité 4
plus grand ou égal	≥	.GE.	de priorité 4
plus grand	>	.GT.	de priorité 4
différent	≠	.NE.	de priorité 4
négation	~	.NO.	de priorité 5
ET	∧	.AND.	de priorité 4
OU	∨	.OR.	de priorité 4



En cas de priorité égale, l'évaluation est supposée se faire de gauche à droite.

Le signe = est également le signe d'affectation.

#### 4. L'opérateur de retard (n)

L'opérateur de retard marque un décalage dans le temps; il peut s'appliquer à une variable ou à une expression; dans ce dernier cas, il s'applique à toutes les variables de l'expression.

L'opérateur de retard s'écrit toujours derrière le nom de la variable ou derrière l'expression sur laquelle il porte; il est de priorité maximum 7, la priorité de niveau 6 étant réservée aux opérateurs économétriques définis au paragraphe suivant.

Ainsi -  $VAR(n) = VAR[I-n]$   
 -  $LOG(VAR)(1) = LOG(VAR[I-1])$   
 -  $(VAR1(1) + LOG(VAR1)VAR2(1))(1)$   
      $= VAR1[I-2] + LOG(VAR1[I-1]/VAR2[I-2])$

#### 5. Les fonctions ou opérateurs spécifiques au problème économétrique.

Ce sont les fonctions de

	<u>pour LEC</u>	<u>pour DBE</u>	
coefficient	<u>C</u>	C :	de priorité 6
différence	<u>D</u>	DIF :	de priorité 6
rapport	<u>R</u>	RAP :	de priorité 6
logarithme	<u>L</u>	LOG :	de priorité 6
absolu	<u>A</u>	ABS :	de priorité 6
average	<u>AV</u>	AV :	de priorité 6
average growth	<u>AG</u>	AG :	de priorité 6
maximum	<u>MX</u>	MAX :	de priorité 6
minimum	<u>MN</u>	MIN :	de priorité 6

Ces fonctions que l'on peut faire précéder d'un paramètre  $n$  peuvent porter sur une variable VAR ou sur une expression (EXP); elles ont alors la signification suivante :

<u>Fonctions LEC</u>	<u>Signification</u>	<u>par défaut, <math>n</math> vaut</u>
$n$ <u>C</u> numérique	C[numérique]	-
$n$ <u>D</u> VAR	VAR[I] - VAR[I-n]	1
$n$ <u>D</u> (EXP)	(EXP) - (EXP) (n) (1)	1
$n$ <u>R</u> VAR	VAR[I] $\div$ VAR[I-n]	1
$n$ <u>R</u> (EXP)	(EXP) $\div$ (EXP) (n)	1
$n$ <u>L</u> VAR	Logarithme (VAR[I])	-
$n$ <u>L</u> (EXP)	Logarithme (EXP)	-
$n$ <u>A</u> VAR	Absolu (VAR[I])	-
$n$ <u>A</u> (EXP)	Absolu (EXP)	-
$n$ <u>AV</u> VAR	$\rho$ VAR $\sum_{i=\rho \text{ VAR}-n+1}^{\rho \text{ VAR}}$ (2) (4)	$\rho$ VAR
$n$ <u>AV</u> (EXP)	(EXP $\rightarrow$ VAR) (3)	$\rho$ VAR
$n$ <u>AG</u> VAR	$\left[ \frac{\text{VAR}[\rho \text{ VAR}]}{\text{VAR}[\rho \text{ VAR}-n]} \right]^{\frac{1}{n}}$	$\rho$ VAR
$n$ <u>AG</u> (EXP)	(EXP $\rightarrow$ VAR)	$\rho$ VAR
$n$ <u>MX</u> VAR	MAX(VAR[I-n+1], ..., VAR[ $\rho$ VAR])	$\rho$ VAR
$n$ <u>MX</u> (EXP)	(EXP $\rightarrow$ VAR)	$\rho$ VAR
$n$ <u>MN</u> VAR	MIN(VAR[I-n+1], ..., VAR[ $\rho$ VAR])	$\rho$ VAR
$n$ <u>MN</u> (EXP)	(EXP $\rightarrow$ VAR)	$\rho$ VAR

(1) Voir l'opérateur de retard.

(2) VAR[ $\rho$ VAR] est l'élément de VAR d'indice le plus élevé.

(3) (EXP  $\rightarrow$  VAR) l'expression doit d'abord être évaluée pour créer une nouvelle série, sur laquelle portera la fonction.

(4) AV, AG, MX, MN se traduiront par l'appel d'une fonction écrite par ailleurs, et qui devra se trouver en mémoire au moment de l'exécution du programme.



En plus des fonctions que nous venons de décrire, nous avons également prévu que des combinaisons de ces fonctions peuvent porter sur une même variable ou sur une même expression.

Ainsi par exemple : DL VAR est la différence première du logarithme de VAR

2 DL VAR est la différence deuxième du logarithme de VAR

RD VAR est le rapport d'ordre 1 de la différence première de VAR

n R m D VAR est le rapport d'ordre n de la différence d'ordre m de VAR

Les différentes combinaisons de fonctions admises (1) et leur signification sont :

<u>pour LEC</u>	<u>pour DBE</u>	<u>Signification</u>
( <u>R</u> VAR-1)	(RAP:VAR-1)	$\frac{\text{VAR}[I] - \text{VAR}[I-1]}{\text{VAR}[I-1]}$
<u>R</u> <u>D</u> VAR	RAP:DIF:VAR	$\frac{\text{VAR}[I] - \text{VAR}[I-1]}{\text{VAR}[I-1] - \text{VAR}[I-2]}$
<u>L</u> <u>D</u> VAR	LOG:DIF:VAR	$\text{LOG}(\text{VAR}[I] - \text{VAR}[I-1])$
( <u>RD</u> VAR-1)	(RAP:DIF:VAR-1)	$\frac{(\text{VAR}[I] - \text{VAR}[I-1]) - (\text{VAR}[I-1] - \text{VAR}[I-2])}{\text{VAR}[I-1] - \text{VAR}[I-2]}$
<u>D</u> <u>R</u> VAR	DIF:RAP:VAR	$\frac{\text{VAR}[I]}{\text{VAR}[I-1]} - \frac{\text{VAR}[I-1]}{\text{VAR}[I-2]}$
<u>L</u> <u>R</u> VAR	LOG:RAP:VAR	$\text{LOG} \left( \frac{\text{VAR}[I]}{\text{VAR}[I-1]} \right)$
<u>D</u> <u>L</u> VAR	DIF:LOG:VAR	$\text{LOG}(\text{VAR}[I]) - \text{LOG}(\text{VAR}[I-1])$

---

(1) Il appartient à l'utilisateur de définir d'autres fonctions ou combinaisons de fonctions. Dans le dossier Utilisateur du Software de Traduction présenté à la fin de ce mémoire, nous définissons comment les introduire dans le programme traducteur.



<u>R</u> <u>L</u> VAR	RAP:LOG:VAR	$\frac{\text{LOG}(\text{VAR}[I])}{\text{LOG}(\text{VAR}[I-1])}$
( <u>RL</u> VAR-1)	(RAP:LOG:VAR-1)	$\frac{\text{LOG}(\text{VAR}[I]) - \text{LOG}(\text{VAR}[I-1])}{\text{LOG}(\text{VAR}[I-1])}$
( <u>DR</u> VAR-1)	(DIF:RAP:VAR-1)	$\frac{\text{VAR}[I] - \text{VAR}[I-1]}{\text{VAR}[I-1]} - \frac{\text{VAR}[I-1] - \text{VAR}[I-2]}{\text{VAR}[I-2]}$
( <u>LR</u> VAR-1)	(LOG:RAP:VAR-1)	$\text{LOG}\left(\frac{\text{VAR}[I] - \text{VAR}[I-1]}{\text{VAR}[I-1]}\right)$

Enfin, en ce qui concerne la présentation d'une équation économétrique, les conventions prises dans IESEM implique que celle-ci soit présentée dans la forme suivante :

la série endogène,  
le signe d'affectation,  
l'ensemble des variables explicatives..

Etant donné la nature économique d'une équation économétrique, nous avons de manière arbitraire limité le nombre des fonctions et combinaisons de fonctions qui peuvent porter sur la variable endogène de l'équation.

Prenons en effet A VAR = la valeur absolue de la variable endogène

ou A D VAR = la valeur absolue de la différence première de la variable endogène;

on aurait perdu les variations de signes de la série d'observations à expliquer et les résultats deviendraient difficiles à interpréter d'un point de vue économique.

Les fonctions admises qui peuvent porter sur la variable endogène sont :

pour LEC

pour DBE

VAR

=

VAR

=

<u>L</u> VAR	=	LOG : VAR	=
<u>D</u> VAR	=	DIF : VAR	=
<u>R</u> VAR	=	RAP : VAR	=
<u>R</u> VAR-1	=	RAP : VAR-1	=
<u>DL</u> VAR	=	DIF : RAP : VAR	=
<u>LD</u> VAR	=	LOG : DIF : VAR	=
<u>DR</u> VAR	=	DIF : RAP : VAR	=
<u>RD</u> VAR	=	RAP : DIF : VAR	=
<u>DR</u> VAR-1	=	DIF:RAP:VAR-1	=
<u>RD</u> VAR-1	=	RAP:DIF:VAR-1	=
<u>LR</u> VAR	=	LOG : RAP : VAR	=
<u>RL</u> VAR	=	RAP : LOG : VAR	=
<u>LR</u> VAR-1	=	LOG:RAP:VAR-1	=
<u>RL</u> VAR-1	=	RAP:LOG:VAR-1	=

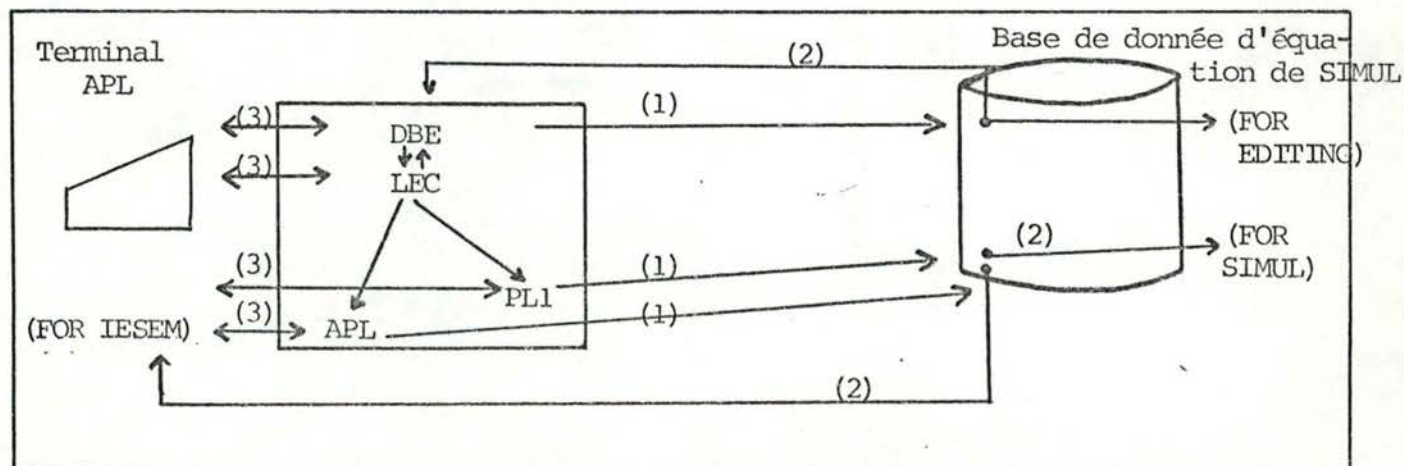
## Chapitre 4 - LE SOFTWARE DE TRADUCTION

Basé sur les formes LEC et DBE des équations économétriques, le software de traduction a pour objectif de générer à une extrémité le littéral APL de l'équation et à l'autre le littéral PL1.

Avant de passer à l'examen de l'algorithme de traduction, nous allons décrire l'organisation générale de ce Software.

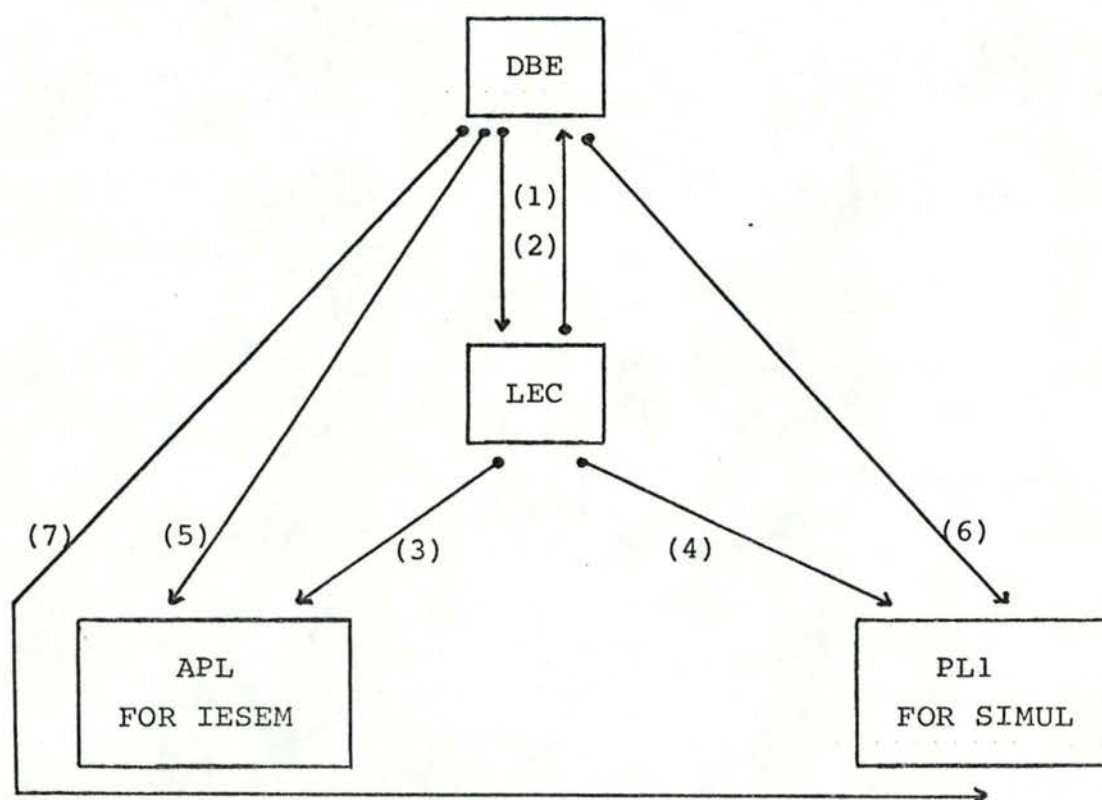
S1 : Son organisation générale

Schématiquement, l'interface réalisé peut se représenter comme suit :



- (1) Les littéraux DBE, PL1 et APL sont stockés dans la base de données d'équations.
- (2) Stockées dans la base de données d'équations, les formes DBE, APL et PL1 sont accessibles au software IESEM pour les unes, SIMUL pour les autres.
- (3) En cours de traduction, les littéraux DBE, LEC, APL et PL1 sont accessibles au terminal APL.

Sept fonctions principales réalisent le software de traduction.



- (1) (2) Deux fonctions LECIN et DBIN permettent de passer d'une forme jumelle à l'autre, la traduction APL ou PL1 se faisant toujours à partir de la forme LEC.
- (3) La fonction APLT permet de traduire la forme LEC en syntaxe APL.
- (4) La fonction PL1T permet de traduire la forme LEC en syntaxe PL1.



(5) (6) Les fonctions TAPL et TPL1 permettent de passer directement de la forme DBE à la forme APL et PL1 en enchaînant respectivement LECIN et APLT, LECIN et PL1T.

(7) TRAD 'EQUATION1, EQUATION2, EQUATION3, ...' est une fonction qui pour un nombre quelconque d'équations, génère et leur forme APL et leur forme PL1.

Chacune de ces fonctions est illustrée dans le DOSSIER UTILISATION DU SOFTWARE, que le lecteur peut trouver à la fin de ce travail.

S2 : Les principes de base du traducteur d'une équation économétrique LEC, dans le langage de programmation APL (fonction (3) : APLT).

Pour résoudre le problème de la traduction d'une équation économétrique LEC dans un langage de programmation donné, nous allons la décomposer en deux parties, la partie gauche qui représente la série à expliquer et la partie droite qui constitue une expression dont les éléments de base sont les variables explicatives.

La traduction de la partie gauche doit se faire conformément aux conventions IESEM et SIMUL, à savoir que la variable endogène doit figurer seule dans la partie gauche; si une fonction LEC porte sur elle, il faut la transformer de façon à la faire apparaître en début de la partie droite :

ainsi l'équation 'L VAR = expression'

doit être traduite en APL par 'VAR[I] ← \* expression'

Pour une description complète de l'algorithme de traduction de la partie gauche nous renvoyons le lecteur au DOSSIER DESCRIPTIF DES FONCTIONS (fonction LEFT) qui se trouve à la fin de ce travail.

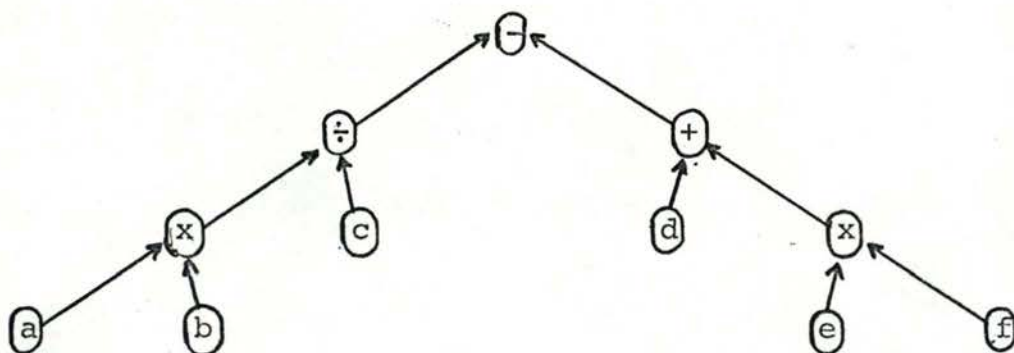
En ce qui concerne la partie droite de l'équation économétrique, nous nous trouvons devant le problème de la traduction d'une

expression arithmétique et/ou booléenne. Il s'agit donc de reconnaître la structure de l'expression et de la traduire dans le langage de programmation approprié.

Une expression est un algorithme qui permet, à l'aide de valeurs connues, de déterminer une valeur inconnue. Cet algorithme peut être représenté par une arborescence dont chaque noeud terminal représente une variable et chaque noeud non terminal une opération :

ainsi, l'expression 'a x b ÷ c - (d + e x f)'

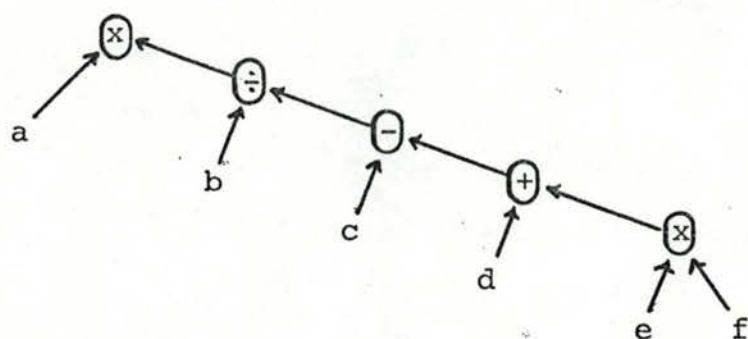
indique un algorithme qui peut être représenté par l'arborescence suivante :



Bien qu'il ne s'agisse pas pour nous d'évaluer une expression mais de la traduire d'une syntaxe dans une autre, il nous faut en reconnaître la structure : la traduction dépend en effet des règles d'évaluation qui diffèrent d'un langage de programmation à l'autre.

Le langage APL qui nous intéresse ici possède seulement une règle d'association à droite. C'est ainsi que si l'on ne modifie pas la syntaxe de l'expression 'a x b ÷ c - (d + e x f)' définie plus haut, on l'évaluera selon une structure d'opérateurs complètement différente, schématisée par l'arborescence suivante :





Lors de la conception du software de traduction, nous devons rechercher un algorithme général capable de générer deux langages de programmation dont les règles d'évaluation sont totalement différentes.

Deux solutions s'offrent à nous pour résoudre ce problème de traduction :

- passer par une forme intermédiaire en notation polonaise et dans une seconde étape générer le langage de programmation désiré,
- parcourir l'expression selon l'ordre de Parry et générer directement le langage de programmation désiré.

A. Dans un premier temps, nous allons montrer comment nous sommes arrivés à préférer l'ordre de Tarry.

B. Dans un second, comment nous l'avons mis en oeuvre dans ses grandes lignes.

A. L'algorithme de traduction en notation polonaise a pour objectif de décomposer une expression de façon telle que les opérandes sur lesquelles portent les opérateurs soient définis de manière unique.

Dans le cas d'opérateurs arithmétiques binaires appartenant à des niveaux de priorité différents, l'algorithme de traduction

---

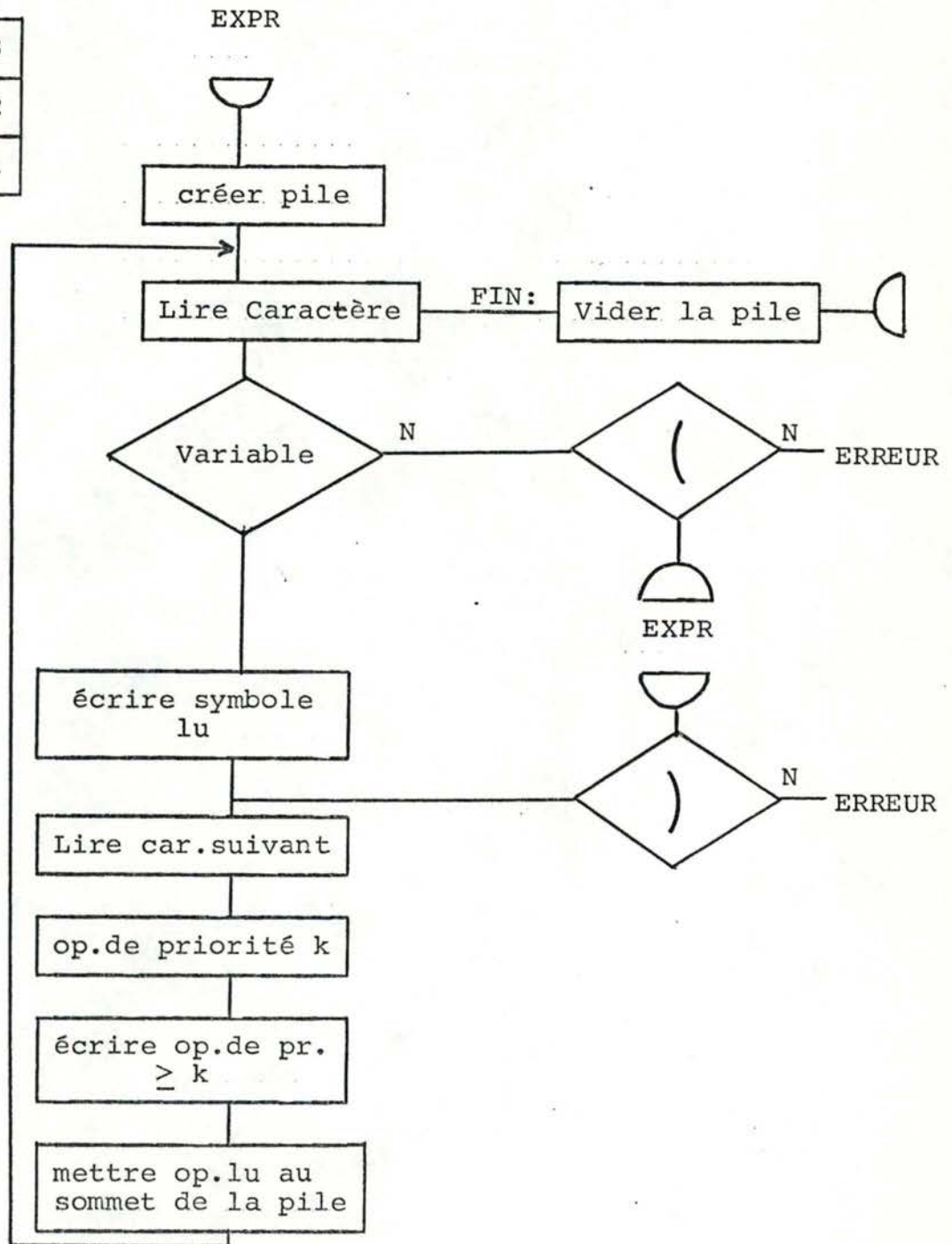
(1) Les concepts d'expression et de notation polonaise, d'arborescence et d'ordre de Tarry sont repris des cours de "Théorie des langages: Technique de compilation" donné par Mr Leroy et de "Théorie des graphes" donné par Mr Fichet aux FNDDP-NAMUR.



en notation polonaise post fixée, se présente comme suit :

Soit

*	3
$x \div$	2
+ -	1



Ainsi l'expression : 'c x (a + b) x k - d \* c'  
 devient en notation polonaise postfixée :  
 'c a b + x k x d e \* -'

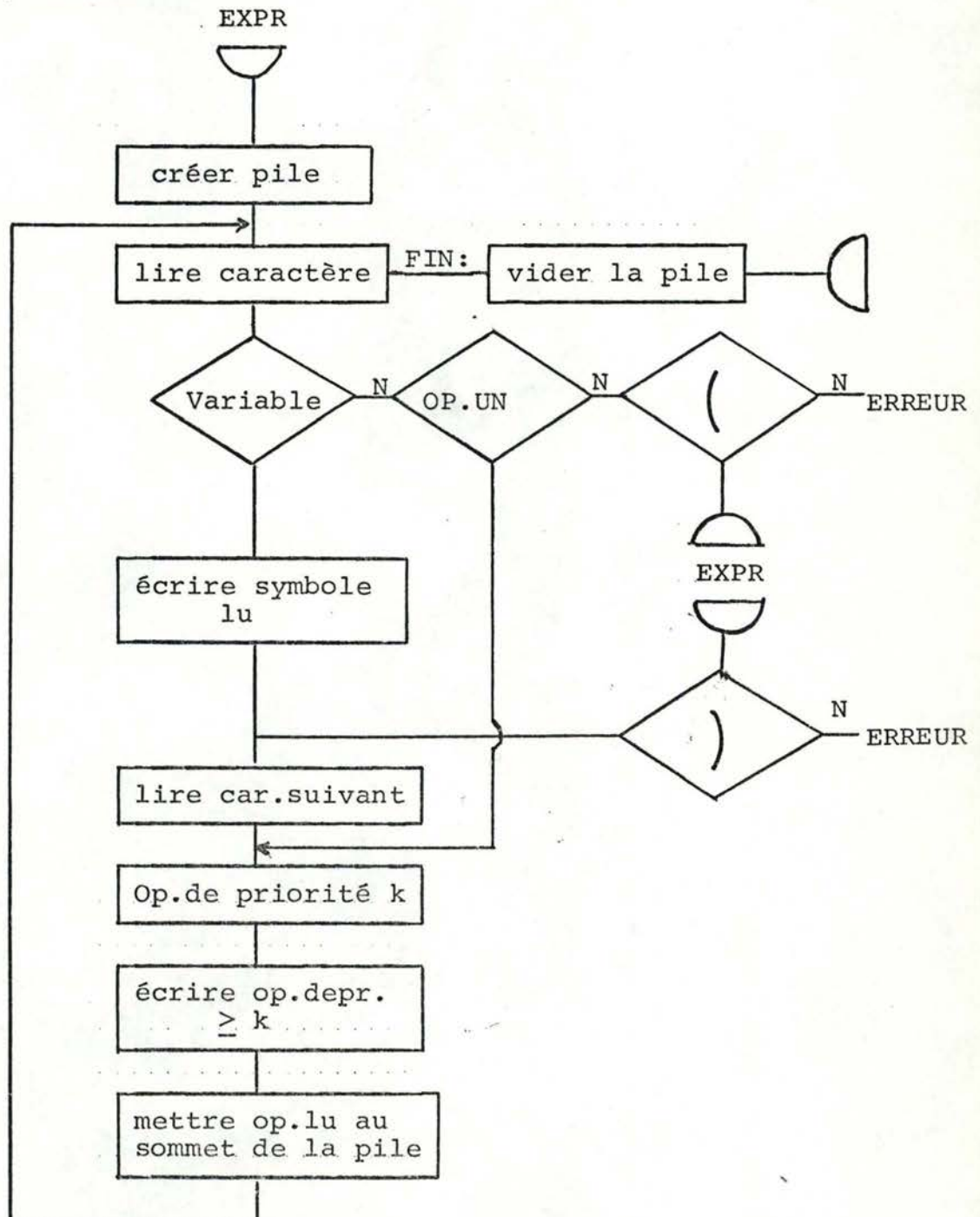
Etendu au cas d'opérateurs unaires, l'algorithme devient :

Soit

OP.RETARD	5
OP.UN	4
*	3
$\times \div$	2
+ -	1

(1)

D, R, ...



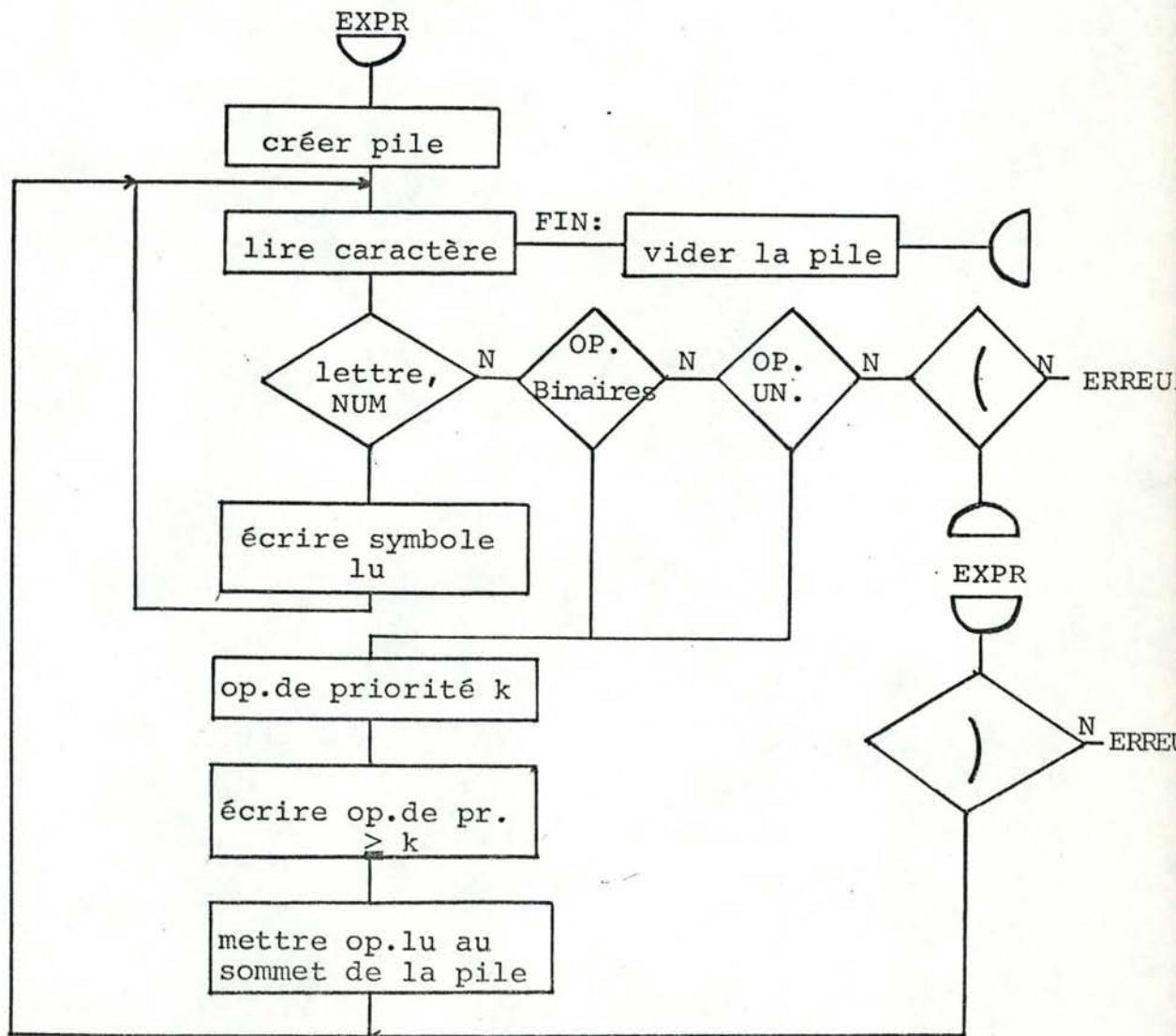
ainsi l'expression : 'c x (a + b) x D k (1) - R (d \* c)'  
devient en notation polonaise postfixée :

'c a b + x k (1) D x d c \* R -'

Appliqué au cas de la traduction d'une expression LEC, en notation polonaise, l'algorithme devient :

Soit

OP.RETARD	5	(1)
OP.UN	4	<u>D</u> , <u>R</u> , ...
*	3	
x $\left(\frac{a}{b}\right)$ /	2	
+ -	1	





ainsi l'expression : ' $\rho 1 \times \underline{D} (\underline{DIT} - \underline{DTG}(1) + \underline{DTC})(1)$ '  
devient en notation polonaise postfixée :

$\rho 1$  (1)  $\underline{D} \times$   
IT  $\underline{D}$  TG (1)  $\underline{D} -$  TC  $\underline{D} +$

Partant de cette forme intermédiaire, la seconde étape de traduction en syntaxe APL devrait se dérouler comme suit :

(1)	(2)
1 $\rho 1$ :.1. C[1]	
2 IT :.2. IT[I]	
3 $\underline{D}$ :.2. IT[I] - IT[I-1]	
4 TG :.3. TG[I]	
5 (1):.3. TG[I-1]	
6 $\underline{D}$ :.3. TG[I-1] - TG[I-2]	
7 - :.2. (IT[I] - IT[I-1]) - TG[I-1] - TG[I-2]	
8 TC :.3. TC[I]	
9 $\underline{D}$ :.3. TC[I] - TC[I-1]	
10 + :.2. (IT[I] - IT[I-1]) - (TG[I-1] - TG[I-2]) - TC[I] - TC[I-1]	
11 (1):.2. (IT[I-1]-IT[I-2])-(TG[I-2]-TG[I-3])-TC[I-1]-TC[I-2]	
12 $\underline{D}$ :.2. ((IT[I-1]-IT[I-2])-(TG[I-2]-TG[I-3])-TC[I-1]-TC[I-2]) -(IT[I-2]-IT[I-3])-(TG[I-3]-TG[I-4])-TC[I-2]-TC[I-3]	
13 x :.1. C[1]x((IT[I-1]-IT[I-2])-(TG[I-2]-TG[I-3])-TC[I-1] -TC[I-2])-(IT[I-2]-IT[I-3])-(TG[I-3]-TG[I-4])-TC[I-2] -TC[I-3]	

$TG[I-1] - TG[I-2]$   
 $IT[I] - IT[I-1]$   
 $C1$

- (1) représente les différents éléments rencontrés dans l'expression en notation polonaise  
(2) représente les hauteurs successives de la pile.

Le premier élément trouvé dans l'expression est C1; il s'agit d'un coefficient et est donc traduit par C[1];

Le second élément trouvé dans l'expression est IT; il s'agit d'une variable et est donc traduit par IT[I];

Le troisième élément trouvé dans l'expression est D; il s'agit de l'opérateur économétrique de différence première; il s'applique à l'élément au sommet de la pile. Ce dernier devient  $IT[I] - IT[I-1]$ ;

Le quatrième élément trouvé dans l'expression est TG; il s'agit d'une variable et est donc traduit par  $TG[I]$ ;

Le cinquième élément trouvé dans l'expression est (1); il s'agit de l'opérateur de retard; il s'applique à l'élément au sommet de la pile. Ce dernier devient  $TG[I-1]$ ;

. . . . .

Le septième élément trouvé dans l'expression est -; il s'agit d'un opérateur binaire; il s'applique aux deux éléments au sommet de la pile; ils deviennent  $(IT[I] - IT[I-1]) - TG[I-1] - TG[I-2]$ . Etant donné la seule règle d'association à droite, il a fallu entourer les deux premiers éléments de parenthèses.

. . . . .

Le dixième élément trouvé dans l'expression est +; il s'agit d'un opérateur binaire; il s'applique aux deux éléments au sommet de la pile; ils deviennent  $(IT[I] - IT[I-1]) - (TG[I-1] - TG[I-2]) - TC[I] - TC[I-1]$ . Etant donné la seule règle d'association à droite, l'opérateur + qui a donné naissance à cette ligne 10 se trouve changé en opérateur -.

. . . . .

Au cours de cette seconde étape de traduction, trois difficultés se présentent :

1. L'application d'opérateurs tels que (1), D, ... nécessite un balayage continu du littéral au sommet de la pile, en vue d'introduire ou modifier un décalage dans l'indice de temps exemple : ligne 11.
2. La seule règle d'association à droite bouleverse les règles d'introduction des parenthèses; celles-ci doivent uniquement



tenir compte de la place de l'élément dans l'expression.

Exemple : ligne 7.

$$(IT[I] - IT[I-1]) - TG[I-1] - TG[I-2]$$

est équivalent à

$$(IT[I] - IT[I-1]) - (TG[I-1] - TG[I-2])$$

mais n'est pas équivalent à

$$IT[I] - IT[I-2] - TG[I-2] - TG[I-2].$$

3. Enfin la seule règle d'association à droite perturbe la liste des opérateurs à introduire.

Exemple : ligne 10 : L'opérateur + qui a donné naissance à cette ligne se trouve changé en opérateur -.

De manière plus générale, la perturbation de la liste des opérateurs à introduire, peut se résoudre de trois façons différentes :

.- par un changement de l'ordre des opérations à effectuer :

ex.:

Arithmétique normale	APL
$8 - 4 + 2 = 6$	$8 - 4 + 2 = 2$ mais $8 + 2 - 4 = 6$
$8/2 + 2 = 6$	$8/2 + 2 = 2$ mais $2 + 8/2 = 6$

.- par un changement de signe :

ex.:

Arithmétique normale	APL
$8 - 4 + 2 = 6$	$8 - 4 + 2 = 2$ mais $8 - 4 - 2 = 6$

.- par l'introduction de parenthèses :

ex.:

Arithmétique normale	APL
$8/2 + 2 = 6$	$8/2 + 2 = 2$ mais $(8/2) + 2 = 6$



Une combinaison de ces méthodes est parfois nécessaire :

ex.:

Arithmétique normale	APL
$8 - 2 - 2 - 2 + 3 = 5$	$8 - 2 - 2 - 2 + 3 = 3$
	mais $8 + 3 - 2 + 2 + 2 = 5$

- B. Nous avons choisi une autre optique de traduction pour résoudre ces trois difficultés : parcourir l'expression selon l'ordre Tarry et générer directement le langage de programmation désiré.

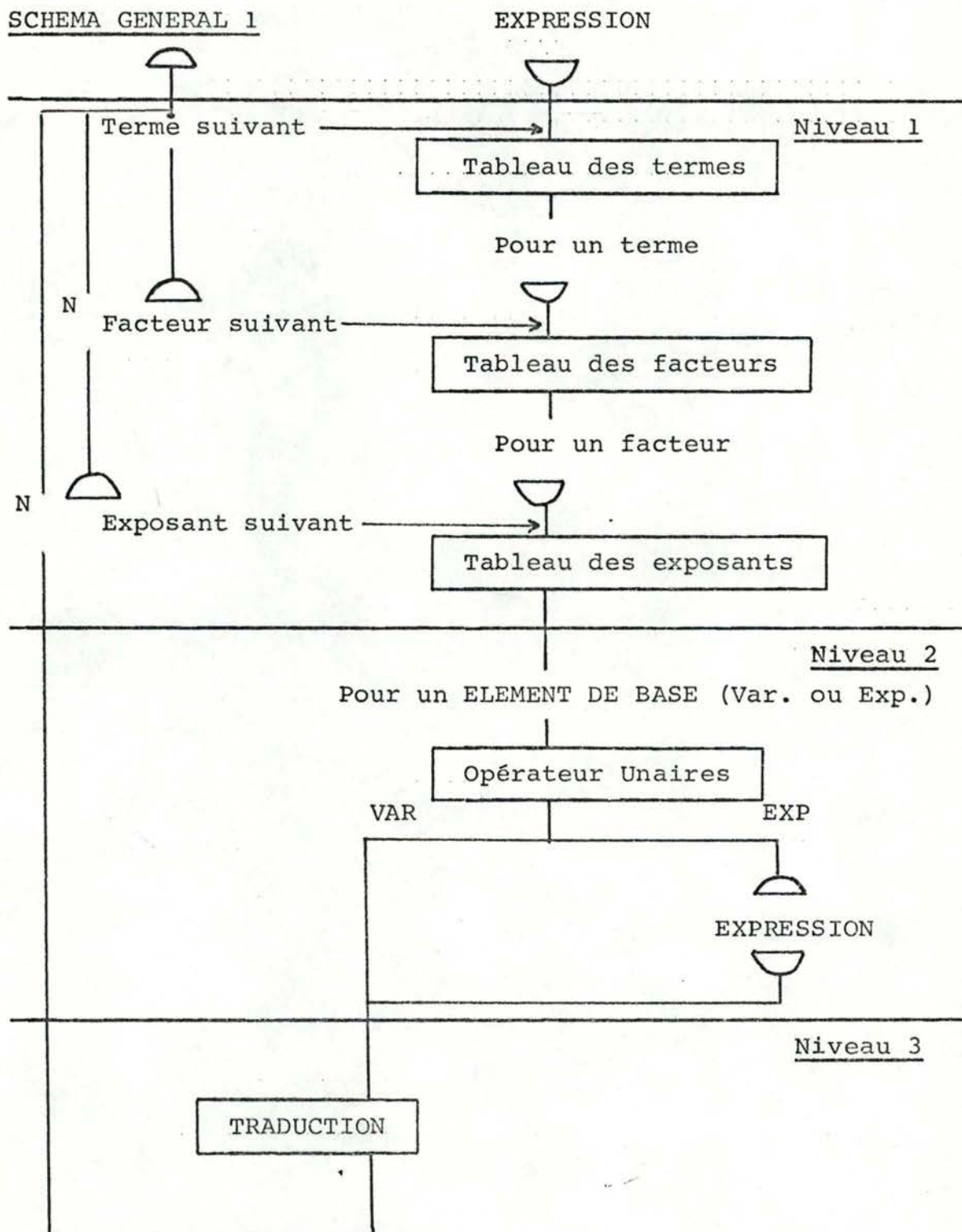
L'ordre de Tarry est celui défini par un promeneur qui, partant de la racine d'une arborescence, rencontre, pour la première fois, chaque sommet en respectant les règles suivantes :

- le promeneur quitte un sommet où il se trouve et chemine dans le sens des arcs, en prenant l'arc le plus à sa droite qu'il n'a pas encore utilisé;
- lorsque cela n'est plus possible, il fait demi-tour et remonte les arcs dans le sens contraire à leur orientation jusqu'à ce qu'il atteigne un sommet d'où parte un arc qu'il n'a pas encore utilisé; ensuite, il applique à nouveau la règle précédente.

Un algorithme en 3 niveaux nous a permis d'atteindre notre objectif de traduction :

- niveau 1 : analyse de la structure des opérateurs binaires :  
la mémorisation de tableaux de termes, facteurs et exposants permet de connaître exactement la place d'un élément dans l'expression.
- niveau 2 : analyse de la structure des opérateurs unaires :  
l'analyse globale d'un élément permet de connaître exactement tous les opérateurs qui portent sur lui.
- niveau 3 : traduction : elle peut se faire directement de manière définitive.

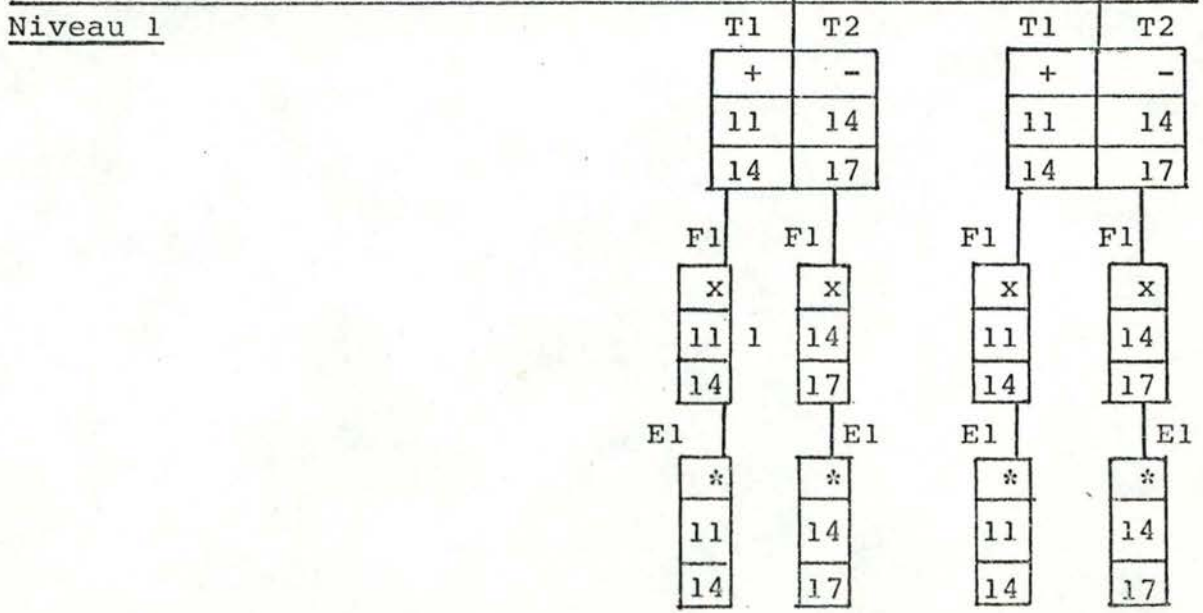
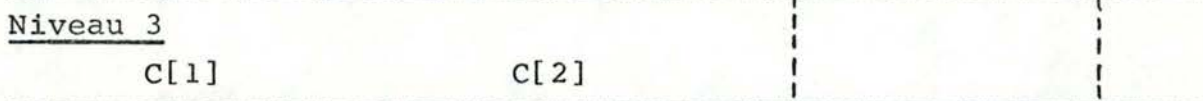
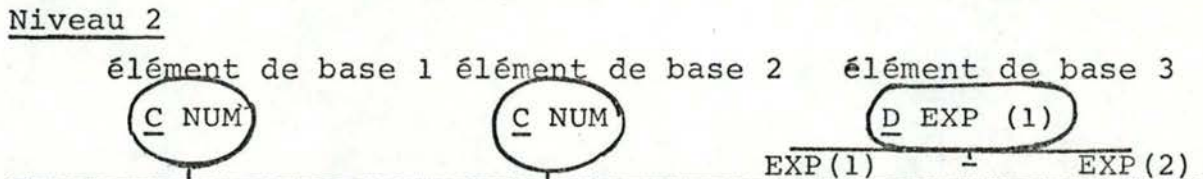
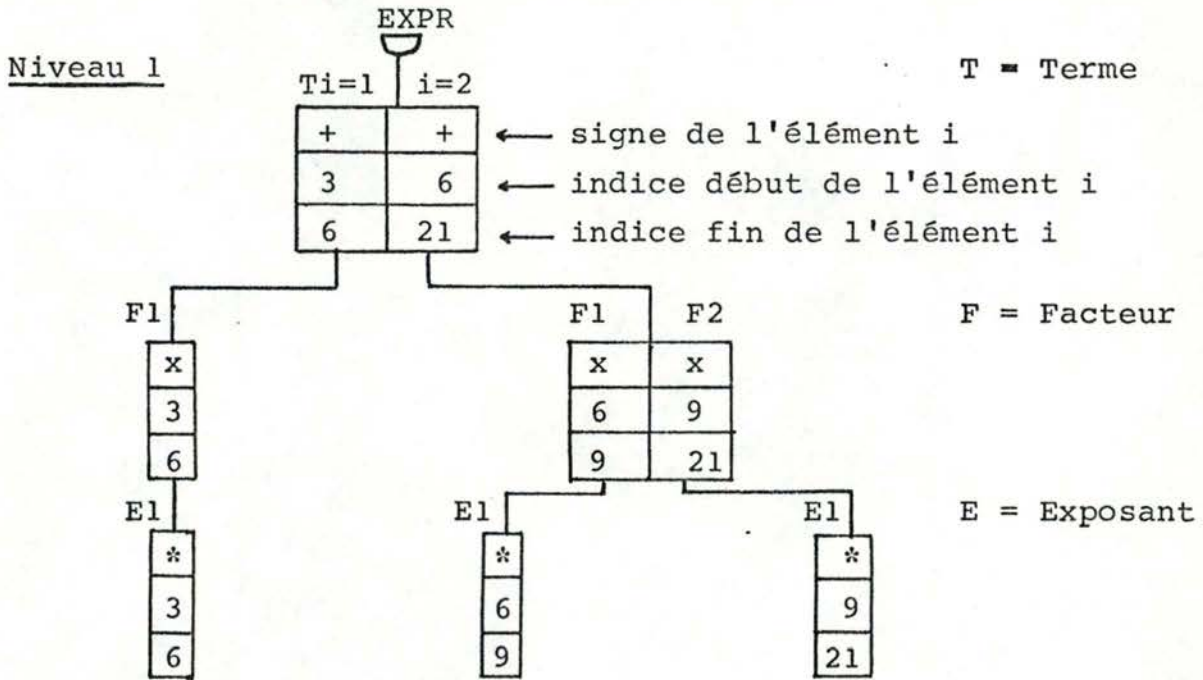
Dans le cas d'expressions arithmétiques, cet algorithme peut se schématiser comme suit :





Exemple : Soit l'expression suivante à traduire :

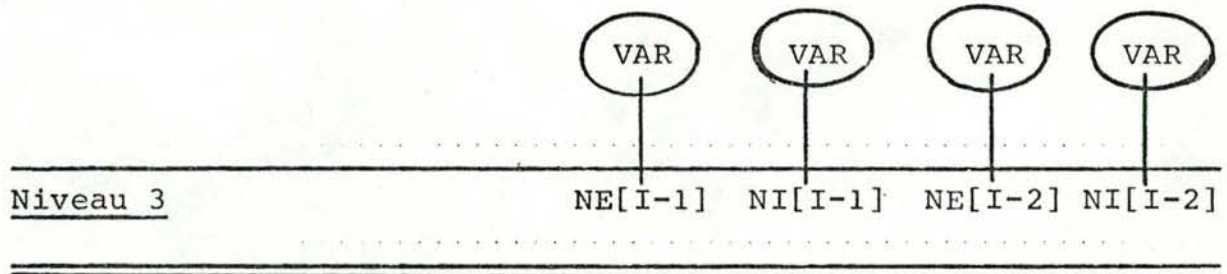
N	A	=	C	1	+	C	2	x	D	(	N	E	-	N	I	)	(	1	)
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20





Niveau 2

élément    élément    élément    élément  
base 31    base 32    base 33    base 34



Arrivé au niveau 3 de traduction, nous devons faire demi-tour et regrouper ces traductions en fonction de la structure des opérateurs binaires rencontrés lors de la décomposition de l'expression. Pour ce faire la seule règle d'association à droite, nous amène à définir trois cas d'introduction de parenthèses. Dans le traducteur, chacun de ces cas est caractérisé par un indicateur particulier.

Il faut introduire des parenthèses dans le cas :

- d'opérateurs unaires (Indicateur SWY)  
ainsi  $\underline{L}$  VAR  
doit être traduit en APL par :  $(@ \text{VAR}[I])$ .
- d'une expression parenthésée (Indicateur SWE)  
ainsi  $(\text{VAR1} + \text{VAR2}) \times \text{VAR3}$   
doit être traduit en APL par :  $(\text{VAR1}[I] + \text{VAR2}[I]) \times \text{VAR3}[I]$ .  
NB :  $\text{VAR3}[I] \times \text{VAR1}[I] + \text{VAR2}[I]$  serait également valable.
- d'opérateurs binaires de priorités différentes (Indicateur SW)  
ainsi  $\text{VAR1} \times \text{VAR2} + \text{VAR3}$   
doit être traduit en APL par :  $(\text{VAR1}[I] \times \text{VAR2}[I]) + \text{VAR3}[I]$   
NB :  $\text{VAR3}[I] + \text{VAR1}[I] \times \text{VAR2}[I]$  serait également valable.

Ces trois indicateurs doivent cependant être remis en question en fonction de la place qu'occupe l'élément traduit dans l'ensemble de l'expression. La seule règle d'association à droite rend en effet facultative l'introduction de parenthèses lorsque l'élément traduit se trouve en fin d'expression:

ainsi :  $\text{VAR1} + \underline{L} \text{VAR2} + \text{VAR3}$

doit être traduit en APL par :  $\text{VAR1}[I] + (@ \text{VAR}[I]) + \text{VAR3}[I]$

mais  $VAR1 + VAR2 + \underline{L} VAR3$

peut être traduit en APL par :  $VAR1[I] + VAR2[I] + \otimes VAR3[I]$

Il faut enfin veiller à ne pas introduire deux fois des parenthèses au même endroit :

ainsi l'expression  $\underline{L} (VAR1 \times VAR2)$

aurait les indicateurs SWY et SWE égaux à 1

et serait traduit en APL par :  $((\otimes VAR1[I] \times VAR2[I]))$

Des exemples détaillés du jeu de ces indicateurs et des principales variables qui permettent de regrouper l'ensemble de la traduction, sont donnés ci-après.

## APLT EQUATION1

LEC FORM:

\*\*\*\*\*

LEC:

VAR = C1 + C2 \* VAR1

(1) (1)

(2)	(2)
C1	1
C1	11
C1	111

111 Y : C[1]  
SWY:0 SWE:0 SW:0  
SA:0

11 YCUM: C[1]  
E : C[1]  
SWY:0 SWE:0 SW:0  
SA:0

1 ECUM: C[1]  
F : C[1]  
SWY:0 SWE:0 SW:0  
SA:0

ECUM: C[1]

C2*VAR1	2
C2	21
C2	211

211 Y : C[2]  
SWY:0 SWE:0 SW:0  
SA:0

21 YCUM: C[2]  
E : C[2]  
SWY:0 SWE:0 SW:0  
SA:0

ECUM: C[2]

VAR1	22
VAR1	221

221 Y : VAR1[I]  
SWY:0 SWE:0 SW:0  
SA:0

22 YCUM: VAR1[I]  
E : VAR1[I]  
SWY:0 SWE:0 SW:0  
SA:0

2 ECUM: C[2]\*VAR1[I]  
F : C[2]\*VAR1[I]  
SWY:0 SWE:0 SW:1  
SA:0

ECUM: C[1]+C[2]\*VAR1[I]

EQ. IN APL FORM FOR NLEST :

\*\*\*\*\*

APL:

VAR[I]\*C[1]+C[2]\*VAR1[I]

VV:

VAR,VAR1

VC:

C[1],C[2]

LMAX:

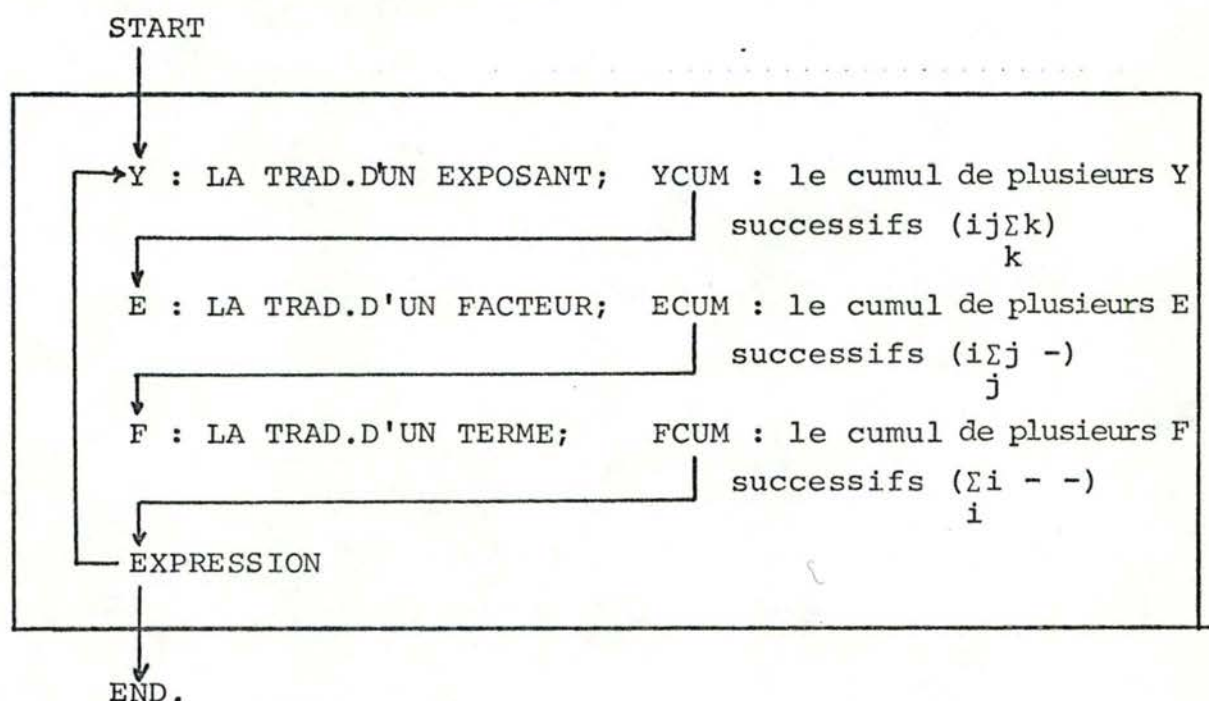
0



- Les colonnes (1) et (2) donnent la correspondance des éléments au moment de leur décomposition selon l'ordre de Tarry (2) et au moment de leur traduction (1).

- 1 indique qu'il s'agit du premier terme de l'expression considérée
- 2 indique qu'il s'agit du second terme de l'expression considérée
- 11 indique qu'il s'agit du premier facteur du premier terme de l'expression considérée
- 111 indique qu'il s'agit du premier exposant du premier facteur du premier terme de l'expression considérée
- 111.1, . . . . etc...

- Les variables utilisées représentent :



- SWY(SA), SWE et SW sont les indicateurs qui vont permettre de déterminer s'il faut ou non introduire des parenthèses en fonction de la place qu'occupe l'élément dans l'expression.

Quelques exemples vont en montrer le fonctionnement.

## 1. Cas de l'indicateur SWY :

LFC FORM:

\*\*\*\*\*

LFC:

$$VAR = C1 + C2 \times DVAP1 \times VAF2$$

		$C1$	1
		$C1$	11
		$C1$	111
111	Y : C[1]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	YCUM: C[1]		
11	E : C[1]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: C[1]		
1	F : C[1]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: C[1]		
		$C2 \times DVAP1 \times VAF2$	2
		$C2$	21
		$C2$	211
211	Y : C[2]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	YCUM: C[2]		
21	E : C[2]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: C[2]		
		$DVAP1$	22
		$DVAP1$	221
221	Y : VAR1[I]-VAR1[I-1]		
	SWY:1 SWE:0 SW:0		
	SA:1		
	YCUM: VAR1[I]-VAR1[I-1]		
22	E : VAR1[I]-VAR1[I-1]		
	SWY:1 SWE:0 SW:0		
	SA:0		
	ECUM: C[2] × (VAR1[I]-VAR1[I-1])		
		VAR2	23
		VAR2	231
231	Y : VAR2[I]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	YCUM: VAR2[I]		
23	E : VAR2[I]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: C[2] × (VAR1[I]-VAR1[I-1]) × VAR2[I]		
2	F : C[2] × (VAR1[I]-VAR1[I-1]) × VAR2[I]		
	SWY:0 SWE:0 SW:1		
	SA:0		
	ECUM: C[1] + C[2] × (VAR1[I]-VAR1[I-1]) × VAR2[I]		

EQ. IN APL FORM FOR NIEST :

\*\*\*\*\*

APL:

VAR[I] ← C[1] + C[2] × (VAR1[I]-VAR1[I-1]) × VAR2[I]

VV:

L'élément 22 D VAR1 traduit par VAR1[I]- VAR1[I-1] met l'indicateur SWY à 1; des parenthèses sont introduites avant de traduire l'élément 23.

L'exemple suivant montre que ces parenthèses sont inutiles lorsque la règle d'association à droite peut jouer.

### APLT EQUATION3

LEC FORM:

\*\*\*\*\*

LEC:

VAR = C1 + C2 × DVAR1

			C1	1
			C1	11
			C1	111
111	Y	: C[1]		
	SWY:0	SWE:0	SW:0	
	SA:0			
	YCUM:	C[1]		
11	E	: C[1]		
	SWY:0	SWE:0	SW:0	
	SA:0			
	ECUM:	C[1]		
1	F	: C[1]		
	SWY:0	SWE:0	SW:0	
	SA:0			
	ECUM:	C[1]		
			C2×DVAR1	2
			C2	21
			C2	211
211	Y	: C[2]		
	SWY:0	SWE:0	SW:0	
	SA:0			
	YCUM:	C[2]		
21	E	: C[2]		
	SWY:0	SWE:0	SW:0	
	SA:0			
	ECUM:	C[2]		
			DVAR1	22
			DVAR1	221
221	Y	: VAR1[I]-VAR1[I-1]		
	SWY:1	SWE:0	SW:0	
	SA:1			
	YCUM:	VAR1[I]-VAR1[I-1]		
22	E	: VAR1[I]-VAR1[I-1]		
	SWY:1	SWE:0	SW:0	
	SA:0			
	ECUM:	C[2]×VAR1[I]-VAR1[I-1]		
2	F	: C[2]×VAR1[I]-VAR1[I-1]		
	SWY:1	SWE:0	SW:1	
	SA:0			
	ECUM:	C[1]+C[2]×VAR1[I]-VAR1[I-1]		
EO. IN API FORM FOR NLEST :				
*****				
API:				
VAR[I]+C[1]+C[2]×VAR1[I]-VAR1[I-1]				



LFC FORM:

\*\*\*\*\*

LFC:

$$VAP = C1 + C2 \times VAR1 + VAR2$$

			C1	1
			C1	11
			C1	111
111	Y	: C[1]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	YCUM: C[1]			
11	E	: C[1]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	ECUM: C[1]			
1	F	: C[1]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	ECUM: C[1]			
			C2×VAR1	2
			C2	21
			C2	211
211	Y	: C[2]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	YCUM: C[2]			
21	E	: C[2]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	ECUM: C[2]			
			VAR1	22
			VAR1	221
221	Y	: VAR1[I]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	YCUM: VAR1[I]			
22	E	: VAR1[I]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	ECUM: C[2]×VAR1[I]			
2	F	: C[2]×VAR1[I]		
	SWY:0	SWE:0 SW:1		
	SA:0			
	ECUM: C[1]+(C[2]×VAR1[I])			
			VAR2	3
			VAR2	31
			VAR2	311
311	Y	: VAR2[I]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	YCUM: VAR2[I]			
31	E	: VAR2[I]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	ECUM: VAR2[I]			
3	F	: VAR2[I]		
	SWY:0	SWE:0 SW:0		
	SA:0			
	ECUM: C[1]+(C[2]×VAR1[I])+VAR2[I]			

EQ. IN APL FORM FOR NLEST :

\*\*\*\*\*

APL:

La traduction de l'élément 2 met l'indicateur SW à 1; en effet, les éléments 21 et 22 doivent être évalués avant l'élément 3.

### 3. Cas de l'indicateur SWE

#### APLT EQUATIONS

LEC FORM:

\*\*\*\*\*

LEC:

$$VAR = C1 + C2 \times (VAR1 - VAR2) \times VAR3$$

			C1	1
			C1	11
			C1	111
111	Y : C[1]			
	SWY:0 SWE:0 SW:0			
	SA:0			
	YCUM: C[1]			
11	E : C[1]			
	SWY:0 SWE:0 SW:0			
	SA:0			
	ECUM: C[1]			
1	F : C[1]			
	SWY:0 SWE:0 SW:0			
	SA:0			
	FCUM: C[1]			
		C2*(VAR1-VAR2)*VAR3	2	
		C2	21	
		C2	211	
211	Y : C[2]			
	SWY:0 SWE:0 SW:0			
	SA:0			
	YCUM: C[2]			
21	E : C[2]			
	SWY:0 SWE:0 SW:0			
	SA:0			
	ECUM: C[2]			
		(VAR1-VAR2)	22	
		(VAR1-VAR2)	221	
		VAR1	221.1	
		VAR1	221.11	
		VAR1	221.111	
221.111	Y : VAR1[I]			
	SWY:0 SWE:0 SW:0			
	SA:0			
	YCUM: VAR1[I]			
221.11	E : VAR1[I]			
	SWY:0 SWE:0 SW:0			
	SA:0			
	ECUM: VAR1[I]			
221.1	F : VAR1[I]			
	SWY:0 SWE:0 SW:0			
	SA:0			
	ECUM: VAR1[I]			
		VAR2	221.2	

		VAR2	221.21
		VAR2	221.211
221.211	Y : VAR2[I]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	YCUM: VAR2[I]		
221.21	E : VAR2[I]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: VAR2[I]		
221.2	F : VAR2[I]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: VAR1[I]-VAR2[I]		
221	Y : VAR1[I]-VAR2[I]		
	SWY:0 SWF:1 SW:0		
	SA:0		
	YCUM: VAR1[I]-VAR2[I]		
22	E : VAR1[I]-VAR2[I]		
	SWY:0 SWE:1 SW:0		
	SA:0		
	ECUM: C[2]*(VAR1[I]-VAR2[I])		
		VAR3	23
		VAR3	231
231	Y : VAR3[I]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	YCUM: VAR3[I]		
23	E : VAR3[I]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: C[2]*(VAR1[I]-VAR2[I])*VAR3[I]		
2	F : C[2]*(VAR1[I]-VAR2[I])*VAR3[I]		
	SWY:0 SWF:0 SW:1		
	SA:0		
	ECUM: C[1]+C[2]*(VAR1[I]-VAR2[I])*VAR3[I]		

EQ. IN APL FORM FOR NLEST :

\*\*\*\*\*

APL:

VAR[I]+C[1]+C[2]\*(VAR1[I]-VAR2[I])\*VAR3[I]

WV:

VAR,VAR1,VAR2,VAR3

VC:

C[1],C[2]

IMAX:

0

La traduction de l'élément 22 met l'indicateur SWE à 1 et introduit des parenthèses dans ECUM.

4. Les différents cas d'introduction de parenthèses en cas d'exposants, sont illustrés par les exemples suivants :



## APLT EQUATION6

LEC FORM:

\*\*\*\*\*

LEC:

$$VAP = C1 + VAR1 * VAR2 * 2$$

		$C1$	1
		$C1$	11
		$C1$	111
111	Y : C[1]		
	SWY:0 SWF:0 SW:0		
	SA:0		
	YCUM: C[1]		
11	F : C[1]		
	SWY:0 SWF:0 SW:0		
	SA:0		
	FCUM: C[1]		
1	F : C[1]		
	SWY:0 SWF:0 SW:0		
	SA:0		
	FCUM: C[1]		
		$VAR1*VAR2*2$	2
		$VAR1*VAR2*2$	21
		$VAR1$	211
211	Y : VAR1[I]		
	SWY:0 SWF:0 SW:0		
	SA:0		
	YCUM: VAR1[I]		
		$VAR2$	212
212	Y : VAR2[I]		
	SWY:0 SWF:0 SW:1		
	SA:0		
21	Y : VAR2[I]		
	YCUM: VAR1[I]*VAR2[I]		
		2	213
213	Y : 2		
	SWY:0 SWF:0 SW:1		
	SA:0		
21	Y : 2		
	YCUM: (VAR1[I]*VAR2[I])*2		
21	F : (VAR1[I]*VAR2[I])*2		
	SWY:0 SWF:0 SW:1		
	SA:0		
	FCUM: (VAR1[I]*VAR2[I])*2		
2	F : (VAR1[I]*VAR2[I])*2		
	SWY:0 SWF:0 SW:0		
	SA:0		
	FCUM: C[1]+(VAR1[I]*VAR2[I])*2		
EQ. IN APL FORM FOR NLEST :			
*****			
APL:			
$VAR[I]+C[1]+(VAR1[I]*VAR2[I])*2$			
VV:			
VAR,VAR1,VAR2			
VC:			
C[1]			
IMAX:			
0			

## APLT EQUATION7

IEC FORM:

\*\*\*\*\*

IEC:

$$VAR = C1 + DVAR1 * VAR2 * 2$$

		C1	1
		C1	11
		C1	111
111	Y : C[1]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	YCUM: C[1]		
11	E : C[1]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: C[1]		
1	F : C[1]		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: C[1]		
		DVAR1*VAR2*2	2
		DVAR1*VAR2*2	21
		DVAR1	211
211	Y : VAR1[I]-VAR1[I-1]		
	SWY:1 SWE:0 SW:0		
	SA:1		
	YCUM: VAR1[I]-VAR1[I-1]		
		VAR2	212
212	Y : VAR2[I]		
	SWY:0 SWE:0 SW:1		
	SA:1		
21	Y : VAR2[I]		
	YCUM: (VAR1[I]-VAR1[I-1])*VAR2[I]		
		2	213
213	Y : 2		
	SWY:0 SWE:0 SW:1		
	SA:1		
21	Y : 2		
	YCUM: ((VAR1[I]-VAR1[I-1])*VAR2[I])*2		
21	E : ((VAR1[I]-VAR1[I-1])*VAR2[I])*2		
	SWY:0 SWE:0 SW:1		
	SA:0		
	ECUM: ((VAR1[I]-VAR1[I-1])*VAR2[I])*2		
2	F : ((VAR1[I]-VAR1[I-1])*VAR2[I])*2		
	SWY:0 SWE:0 SW:0		
	SA:0		
	ECUM: C[1]+((VAR1[I]-VAR1[I-1])*VAR2[I])*2		

EQ. IN APL FORM FOR NLEST :

\*\*\*\*\*

APL:

VAR[I]+C[1]+((VAR1[I]-VAR1[I-1])\*VAR2[I])\*2

VV:

VAR,VAR1,VAR2

VC:

C[1]

IMAX:

1

## APLT EQUATIONS

IEC FORM:

\*\*\*\*\*

IEC:

$$VAR = C1 + DVAR1 * (VAR2 + VAR3) * 2$$

		<u>C1</u>	1
		<u>C1</u>	11
		<u>C1</u>	111
111	Y : C[1] SWY:0 SWE:0 SW:0 SA:0 YCUM: C[1]		
11	E : C[1] SWY:0 SWE:0 SW:0 SA:0 ECUM: C[1]		
1	F : C[1] SWY:0 SWE:0 SW:0 SA:0 FCUM: C[1]		
		<u>DVAR1*(VAR2+VAR3)*2</u>	2
		<u>DVAR1*(VAR2+VAR3)*2</u>	21
		<u>DVAR1</u>	211
211	Y : VAR1[I]-VAR1[I-1] SWY:1 SWE:0 SW:0 SA:1 YCUM: VAR1[I]-VAR1[I-1]		
		(VAR2+VAR3)	212
		VAR2	212.1
		VAR2	212.11
		VAR2	212.111
212.111	Y : VAR2[I] SWY:0 SWE:0 SW:0 SA:0 YCUM: VAR2[I]		
212.11	E : VAR2[I] SWY:0 SWE:0 SW:0 SA:0 ECUM: VAR2[I]		
212.1	F : VAR2[I] SWY:0 SWE:0 SW:0 SA:0 FCUM: VAR2[I]		
		VAR3	212.2
		VAR3	212.21
		VAR3	212.211
212.211	Y : VAR3[I] SWY:0 SWE:0 SW:0 SA:0 YCUM: VAR3[I]		
212.21	F : VAR3[I] SWY:0 SWE:0 SW:0 SA:0 ECUM: VAR3[I]		
212.2	F : VAR3[I] SWY:0 SWE:0 SW:0 SA:0 ECUM: VAR2[I]+VAR3[I]		
212	Y : VAR2[I]+VAR3[I] SWY:0 SWE:1 SW:1 SA:1		
21	Y : VAR2[I]+VAR3[I] YCUM: (VAR1[I]-VAR1[I-1])*VAR2[I]+VAR3[I]		



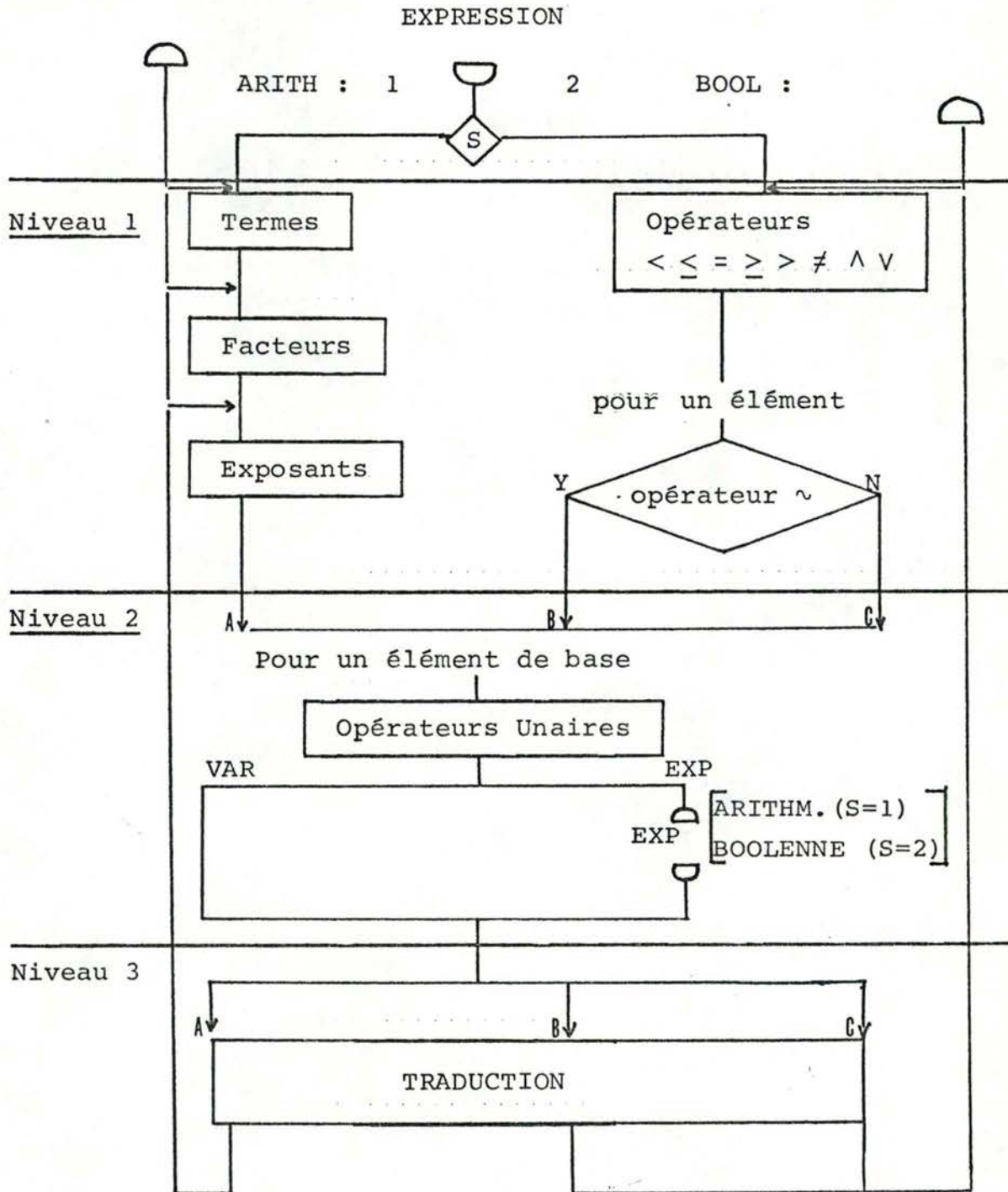
```

213          Y : 2
              SWY:0 SWE:0 SW:1
              SA:1
21          Y : 2
              YCUM: ((VAR1[I]-VAR1[I-1])*VAF2[I]+VAF3[I])*2
21          E : ((VAR1[I]-VAR1[I-1])*VAF2[I]+VAF3[I])*2
              SWY:0 SWF:0 SW:1
              SA:0
              FCUM: ((VAP1[I]-VAP1[I-1])*VAF2[I]+VAF3[I])*2
2          E : ((VAP1[I]-VAP1[I-1])*VAR2[I]+VAF3[I])*2
              SWY:0 SWF:0 SW:0
              SA:0
              FCUM: C[1]+((VAR1[I]-VAR1[I-1])*VAR2[I]+VAF3[I])*2
FO. IN APL FORM FOR NLEST :
*****
APL:
VAP[I]+C[1]+((VAP1[I]-VAP1[I-1])*VAF2[I]+VAF3[I])*2
VV:
VAR,VAP1,VAF2,VAF3
VC:
C[1]
IMAX:
1

```

Etendu aux cas d'expressions arithmétiques et booléennes, et aux opérateurs de comparaison, l'algorithme général 1 devient :

### ALGORITHME GENERAL 2



Arrivé au niveau 3 de traduction, nous devons faire demi-tour et regrouper ces traductions en fonction de la structure des opérateurs booléens rencontrés lors de la décomposition de l'expression. Pour ce faire, la hiérarchie des opérateurs définie dans LEC (1) est prise en compte dans la traduction APL de la façon suivante :

- des parenthèses sont introduites dans le cas d'opérateurs de priorité différente

ainsi l'expression :  $\underline{D}$  VAR2 =  $\underline{D}$  VAR3

doit être traduite en APL par :

$$(VAR2[I] - VAR2[I-1]) = (VAR3[I] - VAR3[I-1])$$

De même l'expression : VAR2 = VAR3

doit être traduite en APL par :  $(\sim VAR2[I]) = VAR3[I]$

- de plus, lorsque les opérateurs booléens sont de même priorité (1), la règle d'association à droite nous oblige à inverser l'ordre de tous les éléments.

Ainsi l'expression : VAR1 = VAR2 V VAR

doit être traduite en APL par :  $VAR[I] V VAR2[I] = VAR1[I]$

- enfin cette seule règle d'association à droite rend toujours facultative l'introduction de parenthèses lorsque l'élément traduit est le dernier élément d'une expression.

Un exemple détaillé de traduction d'une expression booléenne est donné ci-après :

---

(1) Les opérateurs booléens sont tous de même niveau 4, sauf l'opérateur de négation qui est de niveau 5



APLT ' VAR = C1 + C2 \* ( DVAR1 = 0 ) ^ ( VAR2(1) ≥ 100 ) ^ ~VAR3 '

EC FORM:

\*\*\*\*\*

EC:

VAR = C1 + C2 \* ( DVAR1 = 0 ) ^ ( VAR2(1) ≥ 100 ) ^ ~VAR3

	(DVAR1=0)^(VAR2(1)≥100)^~VAR3		
		~VAR3	21
21	Y : (~VAR3[I])		
	SWY:0 SWE:0 SW:0		
	YCUM:(~VAR3[I])		
		(VAR2(1)≥100)	22
	VAR2(1)≥100		
		100	221
221	Y : 100		
	SWY:0 SWE:0 SW:0		
	YCUM:100		
		VAR2(1)	222
222	Y : VAR2[I-1]		
	SWY:0 SWE:0 SW:0		
	YCUM: 100≤VAR2[I-1]		
22	Y : 100≤VAR2[I-1]		
	SWY:0 SWE:1 SW:0		
	YCUM: (~VAR3[I])^(100≤VAR2[I-1])		
		(DVAR1=0)	23
	DVAR1=0		
		0	231
231	Y : 0		
	SWY:0 SWE:0 SW:0		
	YCUM:0		
		DVAR1	232
232	Y : VAR1[I]-VAR1[I-1]		
	SWY:1 SWE:0 SW:0		
	YCUM: 0=VAR1[I]-VAR1[I-1]		
23	Y : 0=VAR1[I]-VAR1[I-1]		
	SWY:1 SWE:1 SW:1		
	YCUM: (~VAR3[I])^(100≤VAR2[I-1])^0=VAR1[I]-VAR1[I-1]		

EO. IN APL FORM FOR NLEST :

\*\*\*\*\*

APL:

VAR[I]+C[1]+C[2]\*(~VAR3[I])^(100≤VAR2[I-1])^0=VAR1[I]-VAR1[I-1]

TV:

VAR,VAR3,VAR2,VAR1

VC:

C[1],C[2]

LMAX:

1

S3 : Le traducteur d'une équation économétrique LEC, dans le langage de programmation PL1. (fonction (4) : PL1T)

Dans la section précédente, nous avons décrit le contexte qui nous a amené à concevoir le programme du traducteur à partir d'une décomposition de l'équation en partie gauche et partie droite, et pour cette dernière, une décomposition selon une structure arborescente. Nous avons décrit les principes de base qui ont guidé son élaboration.

Tournons nous maintenant vers l'autre extrémité de l'interface et examinons le traducteur de la syntaxe LEC vers le langage de programmation PL1.

Nous allons reprendre l'ensemble du traducteur APL. Pour l'adapter à la traduction PL1, l'algorithme général 2 (1) de décomposition selon une structure arborescente peut rester inchangé. Seules la fonction même de traduction et les règles de recombinaison de l'ensemble de la traduction seront différentes.

Pour les expressions arithmétiques, il suffit de respecter les opérateurs binaires de la forme LEC dont les règles d'évaluation sont en effet les mêmes qu'en arithmétique normale.

Le cas d'une expression est comme en APL, mémorisé par l'indicateur SWE, tandis que les opérateurs économétriques sont pris en compte par l'indicateur SWY. Celui-ci prendra une valeur différente en fonction des opérateurs binaires impliqués par l'opérateur économétrique.

Ainsi SWY prend une valeur

- 1 en présence d'un opérateur économétrique de différence
- 2 en présence d'un opérateur économétrique de division
- 3 en présence d'un opérateur économétrique de puissance
- 4 en présence d'une fonction standard (LOG, ...)

---

(1) Voir page 65.

Au moment de regrouper les traductions, la comparaison de cet indicateur avec l'opérateur binaire à introduire, déterminera s'il faut ou non des parenthèses.

Ainsi l'expression  $VAR \times \underline{D} \text{ VAR1}$

aura l'indicateur  $SWY = 1$  ( $\underline{D}$  : opérateur de différence première) et devra être traduite par :  $VAR(I) * (VAR1(I) - VAR1(I-1))$ .

Un exemple complet du jeu de cet indicateur et des principales variables qui permettent de regrouper l'ensemble de la traduction est donné ci-après :

PL1T ' VAR =  $\underline{C1} + \underline{C2} \times \underline{DVAP1} * (VAR2 \times (VAR3 - VAR4))$  '

C FORM:

\*\*\*\*\*

C:

VAR =  $\underline{C1} + \underline{C2} \times \underline{DVAP1} * (VAR2 \times (VAR3 - VAR4))$

$\underline{C1}$	1
$\underline{C1}$	11
$\underline{C1}$	111

11 Y : C(1)  
 SWY:0 SWF:0 SW:0  
 SA:0  
 YCUM: C(1)  
 1 E : C(1)  
 SWY:0 SWF:0 SW:0  
 SA:0  
 ECUM: C(1)  
 F : C(1)  
 SWY:0 SWF:0 SW:0  
 SA:0  
 ECUM: C(1)

$\underline{C2} \times \underline{DVAP1} * (VAR2 \times (VAR3 - VAR4))$	2
$\underline{C2}$	21
$\underline{C2}$	211

11 Y : C(2)  
 SWY:0 SWF:0 SW:0  
 SA:0  
 YCUM: C(2)  
 1 E : C(2)



SWY:0 SWE:0 SW:0  
SA:0  
ECUM:C(2)

70.

DVAR1\*(VAR2\*(VAR3-VAR4))  
DVAR1

22  
221

Y : VAR1(I)-VAR1(I-1)  
SWY:1 SWE:0 SW:0  
SA:1  
YCUM: VAR1(I)-VAR1(I-1)

(VAR2\*(VAR3-VAR4))  
VAR2\*(VAR3-VAR4)  
VAR2  
VAR2

222  
222.1  
222.11  
222.111

Y : VAR2(I)  
SWY:0 SWE:0 SW:0  
SA:0  
YCUM: VAR2(I)  
E : VAR2(I)  
SWY:0 SWE:0 SW:0  
SA:0  
ECUM:VAR2(I)

(VAR3-VAR4)  
(VAR3-VAR4)  
VAR3  
VAR3  
VAR3

222.12  
222.121  
222.121.1  
222.121.11  
222.121.111

Y : VAR3(I)  
SWY:0 SWE:0 SW:0  
SA:0  
YCUM: VAR3(I)  
E : VAR3(I)  
SWY:0 SWE:0 SW:0  
SA:0  
ECUM:VAR3(I)  
F : VAR3(I)  
SWY:0 SWE:0 SW:0  
SA:0  
ECUM:VAR3(I)

VAR4  
VAR4  
VAR4

222.121.2  
222.121.21  
222.121.211

Y : VAR4(I)  
SWY:0 SWE:0 SW:0  
SA:0  
YCUM: VAR4(I)  
E : VAR4(I)  
SWY:0 SWE:0 SW:0  
SA:0  
ECUM:VAR4(I)  
F : VAR4(I)  
SWY:0 SWE:0 SW:0  
SA:0  
ECUM:VAR3(I)-VAR4(I)

Y : (VAR3(I)-VAR4(I))  
SWY:0 SWE:0 SW:0  
SA:0  
YCUM: (VAR3(I)-VAR4(I))  
E : (VAR3(I)-VAR4(I))  
SWY:0 SWE:0 SW:0  
SA:0  
ECUM:VAR2(I)\*(VAR3(I)-VAR4(I))  
F : VAR2(I)\*(VAR3(I)-VAR4(I))  
SWY:0 SWE:0 SW:0  
SA:0  
ECUM:VAR2(I)\*(VAR3(I)-VAR4(I))  
Y : (VAR2(I)\*(VAR3(I)-VAR4(I)))

21  
22.111  
22.11  
22.121.111  
22.121.11  
22.121.1  
22.121.211  
22.121.21  
22.121.2  
22.121  
22.12  
22.1  
2

```

SWY:0 SWF:0 SW:0
SA:0
Y : (VAR2(I)*(VAR3(I)-VAR4(I)))
YCUM: (VAR1(I)-VAR1(I-1))*((VAR2(I)*(VAR3(I)-VAR4(I)))
F : (VAR1(I)-VAR1(I-1))*((VAR2(I)*(VAR3(I)-VAR4(I)))
SWY:0 SWF:0 SW:0
SA:0
FCUM:C(2)*(VAR1(I)-VAR1(I-1))*((VAR2(I)*(VAR3(I)-VAR4(I)))
F : C(2)*(VAR1(I)-VAR1(I-1))*((VAR2(I)*(VAR3(I)-VAR4(I)))
SWY:0 SWF:0 SW:0
SA:0
FCUM:C(1)+C(2)*(VAR1(I)-VAR1(I-1))*((VAR2(I)*(VAR3(I)-VAR4(I)))

```

IN PL1 FORM FOR SIMUL :

\*\*\*\*\*

```

:
(I)=C(1)+C(2)*(VAR1(I)-VAR1(I-1))*((VAR2(I)*(VAR3(I)-VAR4(I)))

```

VAR1,VAR2,VAR3,VAR4

,C(2)

;

Enfin, l'exemple ci-après illustre le cas d'une expression booléenne et celui d'opérateurs de comparaison :

PL1T Z

LEC FORM:

\*\*\*\*\*

LEC:

VAR = C1 + C2\*VAR1\* DA=DB + C3\*VAR2\* T1^(T1vT2)

		<u>DA</u>	21
21	Y : A(I)-A(I-1)		
	SWY:1 SWE:1 SW:0		
	YCUM:(A(I)-A(I-1))		
		<u>DB</u>	22
22	Y : B(I)-B(I-1)		
	SWY:1 SWE:1 SW:0		
	YCUM: (A(I)-A(I-1))=(B(I)-B(I-1))		
		T1	21
21	Y : T1(I)		
	SWY:0 SWE:1 SW:0		
	YCUM:T1(I)		
		~(T1vT2)	22
		T1	22
221	Y : T1(I)		
	SWY:0 SWE:1 SW:0		
	YCUM:T1(I)		
		T2	22
222	Y : T2(I)		
	SWY:0 SWE:1 SW:0		
	YCUM: T1(I)~T2(I)		
22	Y :  ((T1(I)~T2(I)))		
	SWY:0 SWE:0 SW:0		
	YCUM: T1(I)^(T1(I)~T2(I))		

EQ. IN PL1 FORM FOR SIMUL :

\*\*\*\*\*

PI1:

VAR(I)=C(1)+C(2)\*VAR1(I)\*((A(I)-A(I-1))=(B(I)-B(I-1)))+C(3)\*VAR2(I)\*(T1(I)^(T1(I)~T2(I)))

VV:

VAR,VAR1,A,B,VAR2,T1,T2

VC:

C(1),C(2),C(3)

IMAX:

1



Une description complète des fonctions qui composent ce software de traduction est donné dans le DOSSIER DESCRIPTIF DES FONCTIONS que l'on peut trouver à la fin de ce travail.

## CONCLUSIONS

Ce mémoire a pour objectif :

La conception et la réalisation d'un interface, destiné à intégrer les équations économétriques élaborées par un software APL, dans un système plus général de constitution, de simulation, d'archivage et de documentation de modèles économétriques.

L'interface créé peut être étendu dans trois directions :

1. l'élargissement de ces possibilités de traduction.

Le langage LEC peut en effet être étendu à d'autres opérateurs spécifiques à l'économètre; de plus, dans l'estimation jointe de plusieurs équations économétriques, le cas de matrices de coefficients ou de variables est à envisager.

2. l'optimisation du software de traduction.

D'une part, les programmes ont été écrits alors que nous ne connaissions pas toutes les ficelles de l'APL, d'autre part, l'optimisation de la programmation n'était pas l'objectif du travail. Prenons l'espace-mémoire : bon nombre de fonctions du traducteur APL légèrement modifiées pour le cas PL1 ont simplement été dédoublées.

3. l'optimisation de la traduction.

Nous avons vu que l'algorithme de décomposition selon une structure arborescente nous permettait de connaître la place exacte de chaque élément traduit dans l'expression de niveau supérieur. La connaissance de cette place devrait en effet permettre le calcul d'un minimum de parenthèses pour un nombre donné d'opérateurs arithmétiques.

Au terme de ce travail, une voie nouvelle de recherche se dessine :

Un analyseur syntaxique n'aurait plus pour objectif la traduction de chaque opérateur économétrique dans un langage de programmation donné, mais la création de nouvelles séries qui seraient détruites après estimation ou simulation. TESEM et SIMUL travailleraient directement sur une forme LEC adaptée et sur les séries résultantes des opérateurs économétriques.

BIBLIOGRAPHIE - REFERENCES

1. BUREAU DU PLAN : APL in the planing bureau modeling system :  
An interactive estimation and simulation  
package for econometric modeling (IESEM)  
référence : (77)GD-SD/1319/245/ev/1422
2. BUREAU DU PLAN : DOSSIER IESEM  
référence : (77)HS-1491/ev/1617
3. BUREAU DU PLAN : T.S.A.R.  
référence : (77)JW/AB/1373/hg/1488
4. BUREAU DU PLAN : Application SIMUL  
DOSSIER ANALYSE FONCTIONNELLE  
référence (76)DK-HJB/897/ev/942
5. BUREAU DU PLAN : MODELE KUKLOS  
référence (77)GdA/1505/al/1633
6. C.R.I.S.P. : Le Bureau du Plan et le Plan 1971-1975  
C.H. N° 520-521  
Le 7 mai 1971.
7. M. DELEAU et : Méthodes d'analyse des modèles empiriques  
P. MALGRANGE Annales de l'I.A.S.E.E. n° 20 - 1975
8. J. FICHEFET : Théorie des graphes  
Cours donné aux FNOP-NAMUR (1977-1978)
9. W. GILOI : Programmieren in APL  
1. Aufl. Berlin, New-York : de Gruyter 1977  
(de Gruyter Lehrboeh)  
ISBN 3-11-1572-7
10. J. JOHNSTON : Econometric Methods  
(2nd Edition, 1972 - McGraw-Hill)
11. E. KATZAN : APL USER'S GUIDE
12. H. LEROY : Théorie des langages : Techniques de coordination  
Cours donné aux FNDP-NAMUR (1977-1978).



## Annexe 1 : KUKLOS

## Section I - Caractéristiques

Les caractéristiques du modèle peuvent être résumées de la façon suivante :

### 1. Modèle annuel de taille réduite

20 équations  
 4 variables synthétiques  
 2 variables exogènes  
 3 variables instrumentales  
 18 identités  
 10 soldes comptables post récurifs

quelques grands agrégats de la comptabilité nationale Belge ont été choisis de façon à simplifier au maximum tout en permettant de fermer le circuit économique (à l'exclusion des variables exogènes et instrumentales).

### 2. Comportement de longue période sujet à des fluctuations de moyenne période :

une fonction de production permet de calculer une production potentielle et un degré d'utilisation de la capacité endogène; les tendances de l'offre totale et de la demande totale sont générées de façon autonome et la régulation entre les deux s'opère dans la moyenne période par des répercussions localement non linéaires sur les prix, importations, demandes de facteurs de production; toutes les variables restent cependant contraintes par des comportements théoriques explicités satisfaits à long terme.

### 3. Description de la sphère économique réelle; les mécanismes monétaires supposés actifs à court termes sont absents; le niveau général des prix n'est donc pas déterminé à long terme de façon exhaustive et l'illusion monétaire exclue.

x  
x      x

Les variables (1) introduites et les relations qui les déterminent dans le modèle sont énumérées ici selon un regroupement fonctionnel :

#### 1. Les équations :

##### - les équations de demande

CPO : consommation privée à prix constants  
 CG : consommation publique à prix courants  
 IKO : formation brute de capital fixe à prix constants  
 STO : formation de stocks à prix constants  
 XTO : exportations totales à prix constants  
 MTO : importations à prix constants

(1) Notez que les symboles sont constitués à partir de la nomenclature anglaise :

préfixe P- (price) : indice de prix  
 post fixe -O (constant price) : agrégat à prix constants  
 post fixe -R (rate) : taux, ratio.

- les équations de prix

PCP : prix de consommation privée  
 PCG : prix de la consommation publique  
 PIK : prix de la formation de capital fixe  
 PST : déflateur des variations de stock  
 PXT : prix des exportations totales  
 PMT : prix des importations totales

- les équations de revenu

WER : taux de rémunération salariale  
 WIR : taux de rémunération des indépendants  
 WP : revenus de la propriété  
 LI : taux d'intérêt à long terme  
 DP : consommation de capital

- les équations de population et d'emploi

NA : population active  
 NENI : emploi total  
 NI : emploi d'indépendants

2. Les équations auxiliaires de production (variables synthétiques)

YOP : production nationale brute potentielle  
 DUC : taux d'utilisation de la capacité  
 IKOT : formation de capital fixe théorique  
 NENIT : emploi total théorique

3. Les variables exogènes

XWO : exportations totales des pays de la C.E.E. à prix constants  
 PXW : prix des exportations totales des pays de la C.E.E.  
 t : le temps

4. Les instruments fiscaux

THGR : taux de transferts directs nets de l'Etat aux ménages  
 DTCR : taux d'imposition directe sur les entreprises  
 ITR : taux d'imposition indirecte

5. Les identités

- les composantes de la demande finale

CP : consommation privée à prix courants  
     PCP. CPO  
 CGO : consommation publique à prix constants  
     CG / PCG



IK : formation brute de capital fixe à prix courants  
IKO .PIK  
ST : variation des stocks à prix courants  
STO. PST  
XT : exportations totales à prix courants  
XTO . PXT  
MT : importations totales à prix courants  
MTO . PMT

- la demande finale

FY : demande finale à prix courants  
CP + CG + IK + ST + XT  
FYO : demande finale à prix constants  
CPO + CGO + IKO + STO + XTO  
PFY : déflateur de la demande finale  
FY / FYO

- le produit national brut

Y : produit national brut à prix courants  
FY - MT  
YO : produit national brut à prix constants  
FYO - MTO  
PY : déflateur du produit national brut  
Y / YO

- les revenus

WE : rémunérations salariales  
WER . NE  
WI : rémunérations des indépendants  
WIR . NI  
YN : revenu national  
Y - DP - IT  
YD : revenu disponible des ménages  
WE + WI + WP + THG

- l'emploi

NE : emploi salarié  
NENI - NE  
U : chômage  
NA - NE - NI  
UR : taux de chômage  
U / NA

- les transferts fiscaux

THG : transferts directs nets de l'Etat aux ménages  
THGR . YD

DTC : impôts directs sur les entreprises  
 DTCR (YN + DP - WE - WI - WP)  
 IT : impôts indirects  
 ITR . CP

à calculer après résolution du modèle

- les soldes comptables

SH : épargne des ménages  
 YD - CP  
 SW : balance commerciale courante  
 XT - MT  
 SS : surplus résiduel courant des entreprises  
 YN + DP - WE - WI - WP - DTC  
 SG : solde des comptes courants de l'Etat  
 IT + DTC - THG - CG

- les taux de solde

SHR : taux d'épargne des ménages  
 SH / YD  
 SWR : taux de balance commerciale courante  
 SW / Y  
 SSR : taux de surplus résiduel courant des entreprises  
 SS / (YN + DP)  
 SGR : taux de surplus courant du secteur public  
 SG / Y

- productivité et taux de rémunération réelle

YON : productivité moyenne du travail  
 YO / (NE + NI)  
 WRO : taux de rémunération (salaires coûts) réel  
 WER / PCP

L'ordre de ces variables sera revu à l'aide de la procédure de Van Der Gissen; on disposera ainsi du modèle final sous la forme la plus efficiente pour obtenir sa solution par la technique classique de simulation (procédure itérative de Gauss-Seidel).

x  
 x      x

L'écriture d'un modèle de longue période suggère des spécifications fonctionnelles explicites qui tiennent compte de contraintes de cohérence et de convergence. Diverses techniques relativement usuelles ont été utilisées quand celles-ci pouvaient contribuer à simplifier le modèle, tout en assurant une certaine crédibilité et une interprétation économique satisfaisante. Parmi d'autres, il a été fait usage de contraintes d'homogénéité dans le temps, de processus de retardement de Koyck, de contraintes d'additivité, de non-linéarité (circulaire) à brones paramétriques, d'estimation non linéaire dans les paramètres et sous contrainte de paramètres a priori, et d'une relativement stricte application des règles de l'analyse dimensionnelle.

## Section II - Le modèle estimé

### Production potentielle

$$\ln YP = - .023130 + \frac{.64462}{\alpha} [\ln NA - .9 \ln NA_{t-1}] + \frac{.017769}{\beta/2} (\ln IKO + \ln IKO_{t-1}) \\ + \frac{.054894}{\gamma(1-\eta) .043} \text{TREND.t} + \frac{.9}{\eta} \ln YO_{t-1}$$

### Degré d'utilisation des capacités

$$DUC = \frac{YO}{YP} / \frac{YO(1970)}{YP(1970)}$$

### Niveau théorique (désiré) des investissements

$$LIKOT = \frac{1}{\frac{.680158}{1/(\alpha+\beta)}} \left[ \left( \ln YO - \frac{.9}{\eta} \ln YO_{t-1} \right) + \frac{.580158}{\alpha \times \eta} (\ln NA_{t-1} - \ln (NE + NI)_{t-1}) \right. \\ \left. + \frac{.9}{\eta} \ln DUC_{t-1} - \frac{.64462}{\alpha} (\ln PIK(LI + \delta) - \ln \frac{WE + WI}{NE + NI}) - \frac{.054894 \Delta \ln YO \times t}{\gamma(1-\eta)} \right]$$

### Niveau théorique (désiré) de l'emploi total

$$LNENIT = \frac{1}{\frac{.64462}{1/\alpha}} \left[ \left( \ln YO - \frac{.9}{\eta} \ln YO_{t-1} \right) - \frac{.017769}{\beta/2} (\ln IKO + \ln IKO_{t-1}) - \frac{.054894 \Delta \ln YO \times t}{\gamma(1-\eta)} \right] \\ + \frac{.9}{\eta} \ln (NE + NI)_{t-1}$$

### Les variables synthétique sont dérivées de la fonction de production estimée (1)

$$\ln YO = - .023130 + \frac{.64462}{(-.733361) (6.70435)} [\ln (NE + NI) - .9 \ln (NE + NI)_{t-1}] + \frac{.017769}{\beta/2} (\ln IKO + \ln IKO_{t-1}) \\ + \frac{.054894 \Delta \ln YO.t}{(8.25004)} + \frac{.9}{\eta} \ln YO_{t-1}$$

$$S.E. = .006705$$

$$S.C. = .999394$$

$$PER = 1954 - 73$$

$$\bar{Y} = .197886$$

### Consommation privée

$$\ln CPO = \frac{.479290}{(2.57797)} + \frac{.726465}{(6.09973)} \ln \frac{YD}{PCP} + \frac{.043330}{(.385025)} \ln \frac{PCP}{PCP_{t-1}} - \frac{.192099}{(1.29915)} \ln \left( \frac{PCP}{PCP_{t-1}} \right)_{t-1} \\ - \frac{1.02974}{(-2.81251)} [LI - .192099 LI_{t-1}] + \frac{.192099}{(1.29915)} \ln CPO_{t-1}$$

$$S.E. = .007404$$

$$S.C. = .999730$$

$$PER = 1955-74$$

$$\bar{Y} = 6.44847$$

(1) Les t-test sont mis sous les coefficients entre parenthèses



Consommation publique

$$\Delta CG = .429311 \text{ PCG} \cdot \Delta CPO + .656477 \Delta (U.WER)$$

(6.37247)                      (1.82348)

$$+ .064497 [\Delta IT + \Delta THG + \Delta DTC - .429311 \text{ PCG} \cdot \Delta CPO - .656477 \Delta (U.WER)]$$

(.389164)

$$S.E. = 3.68185$$

$$S.C. = .999325$$

$$PER. = 1954-73$$

$$\bar{Y} = 10.5639$$

Exportations totales

$$\ln XTO = -3.08821 + 1.07550 \ln XWO - .169670 \ln \left( \frac{PXT}{PMT} \right)_{t-1}$$

(-25.0737) (72.5351)                      (-.582215)

$$S.E. = .028689$$

$$S.C. = .996779$$

$$PER. = 1954-73$$

$$\bar{Y} = 5.84969$$

Investissements totaux (1)

$$\Delta \ln IKO = .65 \left[ .392983 \Delta \ln IKOT + .607017 \Delta \ln IKOT_{t-1} + .35 \Delta \ln IKO_{t-1} \right]$$

(2.03368)

$$+ .061987 [(DUCS1 - DUCI1) - .35 (DUCS1 - DUCI1)_{t-1}]$$

(.184884)

$$- .169544 \left[ \ln \frac{PMT}{PIK} - .35 \ln \left( \frac{PMT}{PIK} \right)_{t-1} \right] + 1.89503 \left[ \ln \frac{DUC}{DUC} - .35 \ln \left( \frac{DUC}{DUC} \right)_{t-1} \right]$$

(1.35051)

$$+ .202838 \left[ \ln \frac{1 + LI}{PY} - .35 \ln \left( \frac{1 + LI}{PY} \right)_{t-1} \right]$$

(.990408)

$$S.E. = .03795$$

$$S.C. = .999966$$

$$PER = 1957-73$$

$$\bar{Y} = .016681$$

Variations de stocks

$$STO = FYO \left[ .083712 + .068892 \frac{\Delta FYO}{FYO} + .094299 \Delta \ln PFY - .085309 \Delta UC \right]$$

(.625919) (.733823)                      (1.94074)                      (-.611984)

$$S.E. = .003381$$

$$S.C. = .366994$$

$$PER = 1954-73$$

$$\bar{Y} = .005616$$

(1) DUCS1, DUCI1 sont des fonctions non linéaires circulaires de DUC

(77) DG.1505/al/1633

Importations totales

$$\Delta \text{MTO} = [.733801 + .693209 (\text{DUCS2} - \text{DUCI2}) + .017124 t] \\ (1.23758) \quad (.772089) \quad (.491906)$$

$$. [.3\Delta \text{CPO} + .45\Delta \text{IKO} + .7\Delta \text{STO} + .4\Delta \text{XTO}] - .020756 \left( \ln \frac{\text{PMT}}{\text{PY}} \text{MTO}_{t-1} \right) \\ (-.089414)$$

$$\text{S.E.} = 11.5250$$

$$\text{S.C.} = .999363$$

$$\text{PER} = 1954-73$$

$$\bar{Y} = 32.1600$$

L'indice des prix à la consommation

$$\Delta \ln \text{PCP} = \Delta \text{ITR} + .213011 + .663303 (\Delta \ln \text{PCP}_{t-1} - \Delta \text{ITR}_{t-1}) \\ (3.01432)$$

$$+ .336697 \left[ \frac{\text{WE} + \text{WI}}{\text{FY}} \cdot \ln \frac{\text{WE} + \text{WI}}{\text{YO}} + \frac{\text{MT}}{\text{FY}} \cdot \ln \text{PMT} + \left( 1 - \frac{\text{WE} + \text{WI} + \text{MT}}{\text{FY}} \right) \ln \text{PIK} (\text{LI} + \delta) - (\ln \text{PCP}_{t-1} - \text{ITR}_{t-1}) \right] \\ (2.99777)$$

$$+ .082204 (\ln \text{DUC} - .663303 \ln \text{DUC}_{t-1}) \\ (.218934)$$

$$\text{S.E.} = .022085$$

$$\text{S.C.} = .984540$$

$$\text{PER.} = 1955-73$$

$$\bar{Y} = .002467$$

L'indice des prix à l'investissement

$$\Delta \ln \text{PIK} = .120642 + .3 \left( .55 \ln \frac{\text{WE} + \text{WI}}{\text{YO}} + .45 \ln \text{PMT} - \ln \text{PIK}_{t-1} \right) + .157441 (\ln \text{DUC} - .7 \ln \text{DUC}_{t-1}) \\ (12.5901) \quad (.233399)$$

$$\text{S.E.} = .040608$$

$$\text{S.C.} = .965446$$

$$\text{PER.} = 1955-73$$

$$\bar{Y} = .120119$$

Déflateur de la consommation publique

$$\ln \text{PCG} = .054880 + .837899 (\ln \text{PCG}_{t-1} + \Delta \ln \text{PCG}_{t-1}) \\ (.245919)$$

$$+ .162101 [.202272 \ln \text{WER} + .779469 \ln \text{PCP} + .018259 \ln (\text{LI} + 1)] \\ (1.89495) \quad (.221343) \quad (.336298)$$

$$\text{S.E.} = .073772$$

$$\text{S.C.} = .996509$$

$$\text{PER} = 1955-73$$

$$\bar{Y} = .004099$$

Déflateur implicite des variations de stocks

$$\ln \text{PST} = .052117 - .06976(\ln \text{PST}_{t-1} + \Delta \ln \text{PST}_{t-1}) + 1.06976[.660025 \ln \text{PIK} + .339975 \ln \text{PMT}]$$

(.858593) (10.9441) (0.80440)

S.E. = .192284

S.C. = .526456

PER. = 1955-74

$\bar{Y}$  = .004168

Prix à l'importation

$$\ln \text{PMT} = - .008110 + 1.07249 \ln \text{PXW}$$

(-.493587) (9.05646)

S.E. = .046763

S.C. = .811918

PER = 1953-73

$\bar{Y}$  = -.124747

Prix à l'exportation

$$\ln \text{PXT} = .090918 + .507943(\ln \text{PXT}_{t-1} + \Delta \ln \text{PXT}_{t-1})$$

(.871426)

$$+ .492057 \left[ .828710 \ln \text{PXW} + .171290 \left( \frac{\text{WE} + \text{WI}}{\text{Y}} \ln \frac{\text{WE} + \text{WI}}{\text{YO}} + \left( 1 - \frac{\text{WE} + \text{WI}}{\text{Y}} \right) \ln \text{PIK} (\text{LI} + \delta) \right) \right]$$

(3.00433) (5.14886)

$$+ .718334(\ln \text{DUC} - .507943 \ln \text{DUC}_{t-1})$$

(1.02229)

S.E. = .033112

S.C. = .922656

PER = 1955-73

$\bar{Y}$  = .006719

Taux de rémunération salariale (1)

$$\ln \text{WER} = - .242107 + .609492(\ln \text{WER}_{t-1} + \Delta \ln \text{WER}_{t-1})$$

(-.804277)

$$+ .390508 \left[ .850630 \ln \frac{\text{YO}}{\text{NE} + \text{NI}} + 1.36893 \ln \text{PCP} \right]$$

(2.27035) (0.69906) (2.16815)

$$- .93463 (\text{UNAS}_{t-1} - .609492 \text{UNAS}_{t-2})$$

(-.958914)

S.E. = .016342

S.C. = .998772

PER = 1955-73

$\bar{Y}$  = .005367

(1) UNAS est une fonction non linéaire circulaire de U



Taux de rémunération des indépendants

$$\Delta \ln \text{WIR} = .038468 + .452362 (\ln \text{WER} - \ln \text{WIR}_{t-1}) + .547638 \Delta \ln \text{WIR}_{t-1} \\ + .652441 \left[ \frac{Y-IT-WE}{FY} - .547638 \left( \frac{Y-IT-WE}{FY} \right)_{t-1} \right] + .619683 \Delta \ln \text{DUC} - .547638 \Delta \ln \text{DUC}_{t-1} \\ (.548607) (3.37108) \quad (1.25153) \quad (1.52899)$$

$$\text{S.E.} = .031176 \quad \text{S.C.} = .994447 \quad \text{PER.} = 1956-73$$

$$\bar{Y} = .003309$$

Taux d'intérêt à long terme

$$\text{LI} = .018128 - .3 \frac{YD-CP+XT-MT+DP+YN-WE-WI-WP-CG+IT-THG}{FY} \\ (.494555) \\ + .125669 \Delta \ln (\text{FYO}-\text{IKO}-\text{STO}) + .543386 \frac{IK+ST}{FY} + .165447 \Delta \ln \text{PFY} \\ (.790297) \quad (2.15543) \quad (1.63763)$$

$$\text{S.E.} = .013323 \quad \text{S.C.} = .478061 \quad \text{PER.} = 1954-74$$

$$\bar{Y} = .112327$$

Emploi total

$$\Delta \ln (\text{NE}+\text{NI}) = .023052 + .590525 [.468764 \text{LNENIT} + .531236 \text{LNENIT}_{t-1} - \ln (\text{NE}+\text{NI})_{t-1}] \\ (5.52218) (5.86858) (1.73885) \\ + .409475 \Delta \ln (\text{NE}+\text{NI})_{t-1}$$

$$\text{S.E.} = .006353 \quad \text{S.C.} = .966888 \quad \text{PER.} = 1955-73$$

$$\bar{Y} = .000293$$

Emploi indépendant

$$\ln \frac{NI}{NE+NI} = -.022954 + \ln \left( \frac{NI}{NE+NI} \right)_{t-1} + .094844 \Delta \ln \text{DUC} \\ (-8.24994) \quad (5.33043)$$

$$\text{S.E.} = .012115 \quad \text{S.C.} = .991569 \quad \text{PER.} = 1955-73$$

$$\bar{Y} = -.022837$$

Offre de main-d'oeuvre

$$\ln NA = 1.84795 + .038389 \ln \frac{WE+WI}{(NE+NI)PCP}$$

(2.48089) (4.37662)

$$- .562556 \left[ \frac{U}{NA} - .784545 \left( \frac{U}{NA} \right)_{t-1} \right] + .784545 \ln NA_{t-1}$$

(-2.23801) (8.84559)

S.E. = .004807

S.C. = .976727

PER. = 1954-73

$\bar{Y} = 8.23299$

Consommation de capital fixe

$$\ln DP = - .207857 + .809243 \ln DP_{t-1} + .190757 \ln PIK (IKO + IKO_{t-1})$$

(-3.63746) (4.98711)

$$+ 1.85338 \left[ \frac{YN-WE-WI-IT-DTC}{FY} - .809243 \left( \frac{YN-WE-WI-IT-DTC}{FY} \right)_{t-1} \right]$$

(2.44546)

S.E. = .018788

S.C. = .998516

PER. = 1954-74

$\bar{Y} = .075174$

Revenu de la propriété

$$\ln WP = .047375 + .9 \ln WP_{t-1} + .107118 \ln (YD-CP) + 1.27316 (LI - .9 LI_{t-1})$$

(.757239) (7.27161) (.793310)

S.E. = .045413

S.C. = .991349

PER. = 1954-74

$\bar{Y} = .522437$

DOSSIER UTILISATEUR

DU

SOFTWARE DE TRADUCTION

1. Structure du Software	DU. 1
2. TAPLGP et TPL1GP	DU. 3
3. TRADGP	DU. 7
4. Quelques caractéristiques	DU.11
5. Introduction de nouvelles fonctions ou combinaisons de fonctions économétriques	DU.12



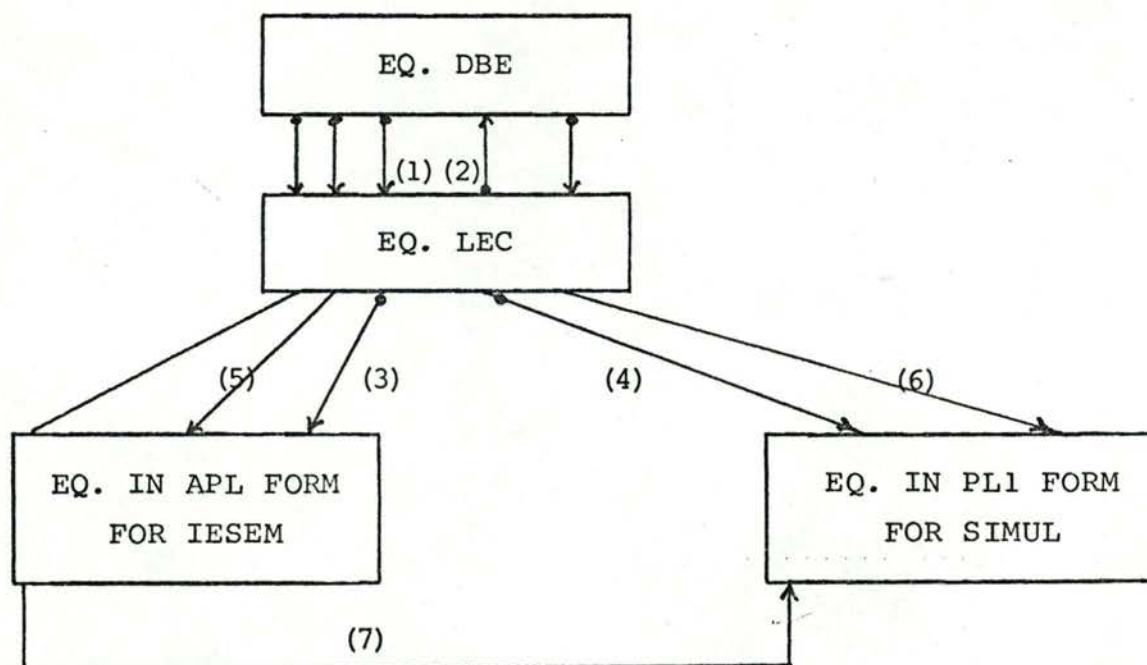
## 1. STRUCTURE DU SOFTWARE

Basé sur le Langage Econométrique Condensé (LEC) défini dans le chapitre 3 du travail qui précède, ce software réalise la traduction d'une équation économétrique LEC dans les langages de programmation APL et PL1.

Les concepts de base sont :

- une équation LEC : équation économétrique écrite dans sa forme LEC
- une équation DBE : équation économétrique écrite dans sa forme DBE; cette forme est jumelle à la forme LEC et est destinée à être stockée dans une Base de donnée d'Equations. Elle est également définie dans le chapitre 3 du travail qui précède.

On peut schématiser le software de la façon suivante :



- (1) : Fonction LECIN : traduit la forme DBE dans sa forme LEC
- (2) : Fonction DBIN : traduit une forme LEC dans sa forme DBE
- (3) : Fonction APLT : traduit une équation LEC en APL
- (4) : Fonction PL1T : traduit une équation LEC en PL1
- (5) : Fonction TAPL : traduit une équation DBE en APL
- (6) : Fonction TPL1 : traduit une équation DBE en PL1
- (7) : Fonction TRAD : traduit une ou plusieurs équations DBE en APL et PL1.

Toutes ces fonctions sont sauvées dans la workspace APL : 100141 NAVYTRAD. Elles y sont groupées de la façon suivante :

- les fonctions (1) (2) (3) et (5) font partie du groupe APL : TAPLGP
- les fonctions (1) (2) (4) et (6) font partie du groupe PL1 : TPL1GP
- les fonctions (1) (2) et (7) font partie du groupe particulier TRADGP.

Les opérations chronologiques pour pouvoir utiliser un groupe de fonctions, sont :

```

)      : n° d'entrée dans le système APL
) SYMBOLS 999
) COPY 100141 NAVYTRAD { TAPLGP
                        { TPL1GP
                        { TRADGP
                        { _____ (1)

```

---

(1) dans ce cas les fonctions de n'importe quel groupe peuvent être utilisées.

## 2. TAPLGP et TPL1GP

En plus des fonctions principales mentionnées plus haut, chaque groupe comprend toutes les fonctions nécessaires à la traduction d'une équation LEC ou DBE en APL ou PL1.

Ce sont, pour TAPLGP :

)FNS

<u>PLT</u>	<u>DEJN</u>	<u>LEGIN</u>	<u>TAPL</u>	<u>AG</u>	<u>ANALYSE</u>	<u>ANYP</u>	<u>APLPOL</u>	<u>APIEL</u>	<u>AV</u>
<u>COL</u>	<u>CONSTRUCT</u>		<u>EXPP</u>	<u>ELTPE</u>	<u>ET</u>	<u>INIT</u>	<u>INITIAL</u>	<u>LAGGED</u>	<u>LEFT</u>
<u>PT</u>	<u>MY</u>		<u>RANGE</u>	<u>TABEAU</u>	<u>TTE</u>				

et pour TPL1GP :

)FNS

<u>PLT</u>	<u>LEGIN</u>	<u>PL1T</u>	<u>TPL1</u>	<u>ANALYSE1</u>	<u>APIEL</u>	<u>ECOL1</u>	<u>CONSTRUCT</u>
<u>EXPP1</u>	<u>ELTPE</u>	<u>ET1</u>	<u>INIT</u>	<u>INITIAL1</u>	<u>LAGGED</u>	<u>LEFT1</u>	<u>LEP1</u> <u>TABEAU</u>
<u>PT1</u>							

Les fonctions non soulignées sont directement utilisables par l'utilisateur.

Leur mode d'emploi est le suivant :

RES ← LEGIN NOMDBE

où NOMDBE est le nom d'une équation dans sa forme DBE

RES est le nom du vecteur contenant le résultat de la traduction de NOMDBE en forme LEC

réalise la traduction d'une équation DBE en syntaxe LEC.

La fonction est a output explicite (RES).



NOMDPE

$$G:YP = C:1 + C:2 * 2AV:IKO + .1 * C:3 * TREND * TEMPS + .9 * LOG:YP(1)$$

RES← LECIN NOMDBE

RES

$$P = \underline{C}1 + \underline{C}2 * 2\underline{AV}IKO + .1 * \underline{C}3 * TREND * TEMPS + .9 * \underline{L}YP(1)$$

RES ← DBIN NOMLEC

réalise juste l'opération inverse à LECIN.

NOMLEC

$$= \underline{C}1 + \underline{C}2 * \underline{L}((WE + WI) \div (NE + NI) * PCP) + \underline{C}3 * (U \div POP - \underline{C}4 * (U \div POP)(1)) + \underline{C}4 * (NA \div POP)(1)$$

RES← DBIN NOMLEC

RES

$$= C:1 + C:2 * LOG:((WE + WI) / (NE + NI) * PCP) + C:3 * (U / POP - C:4 * (U / POP)(1)) + C:4 * (NA / POP)(1)$$

APLT NOMLEC

PL1T NOMLEC

où NOMLEC est le nom d'une équation dans sa forme LEC  
réalise la traduction de l'équation NOMLEC en APL pour APLT  
en PL1 pour PL1T

La fonction est à output implicite et imprime successivement

- la forme LEC : (LEC FORM :)
- la forme APL : (EQ. IN APL FORM FOR NLEST :) pour APLT
- la forme PL1 : (EQ. IN PL1 FORM FOR SIMUL :) pour PL1T

- VV : le vecteur contenant le nom des variables rencontrées au cours de la traduction, la première étant la variable endogène.
- VC : le vecteur contenant la liste des coefficients rencontrés dans l'équation
- LMAX : le retard maximum rencontré lors de la traduction.

Tous ces outputs sont présentés sous la forme d'un vecteur alpha-numérique de même nom :

#### NOMLEC

$$NA = \underline{C1} + \underline{C2} \times \underline{L}((WE + WI) \div (NE + NI) \times PCP) + \underline{C3} \times (U \div POP - \underline{C4} \times (U \div POP)(1)) + \underline{C4} \times (NA \div POP)(1)$$

#### APLT NOMLEC

#### LEC FORM:

\*\*\*\*\*

#### LEC:

$$NA = \underline{C1} + \underline{C2} \times \underline{L}((WE + WI) \div (NE + NI) \times PCP) + \underline{C3} \times (U \div POP - \underline{C4} \times (U \div POP)(1)) + \underline{C4} \times (NA \div POP)(1)$$

#### EQ. IN APL FORM FOR NLEST :

\*\*\*\*\*

#### APL:

$$NA[I] + C[1] + (C[2] \times ((WE[I] + WI[I]) \div (NE[I] + NI[I]) \times PCP[I]) + C[3] \times (U[I] \div POP[I]) - C[4] \times U[I - 1] \div POP[I - 1]) + C[4] \times NA[I - 1] \div POP[I - 1]$$

#### VV:

NA,WE,WI,PCP,NE,NI,U,POP

#### VC:

C[1],C[2],C[3],C[4]

#### LMAX:

1

#### PL1T NOMLEC

#### LEC FORM:

\*\*\*\*\*

#### LEC:

$$NA = \underline{C1} + \underline{C2} \times \underline{L}((WE + WI) \div (NE + NI) \times PCP) + \underline{C3} \times (U \div POP - \underline{C4} \times (U \div POP)(1)) + \underline{C4} \times (NA \div POP)(1)$$

#### EQ. IN PL1 FORM FOR SIMUL :

\*\*\*\*\*

#### PL1:

$$NA(I) = C(1) + C(2) * LOG((WE(I) + WI(I)) / (NE(I) + NI(I)) * PCP(I)) + C(3) * (U(I) / POP(I) - C(4) * (U(I - 1) / POP(I - 1))) + C(4) * (NA(I - 1) / POP(I - 1))$$

#### VV:

NA,WE,WI,NE,NI,PCP,U,POP

#### VC:

C(1),C(2),C(3),C(4)

#### LMAX:

1

TAPL NOMDBE

TPL1 NOMDBE

où NOMDBE est le nom d'une équation dans sa forme DBE.  
 réalise la traduction de l'équation NOMDBE en APL pour TAPL  
 en PL1 pour TPL1

La fonction est à output implicite et imprime successivement

- la forme DBE : (EQ.TO TRADUCT :)
- la forme LEC : (LEC FORM :)
- la forme APL : (EQ.IN APL FORM FOR NLEST :) pour TAPL
- la forme PL1 : (EQ.IN PL1 FORM FOR SIMUL :) pour TPL1
- VV : le vecteur contenant le nom des variables rencontrées au cours de la traduction, la première étant la variable endogène
- VC : le vecteur contenant la liste des coefficients rencontrés dans l'équation
- LMAX : le retard maximum rencontré lors de la traduction

Tous ces outputs sont présentés sous la forme d'un vecteur alphanumérique de même nom.

NOMDBE

$$\text{LOG:YP} = \text{C:1} + \text{C:2} * 2\text{AV:IKO} + .1 * \text{C:3} * \text{TREND} * \text{TEMPS} + .9 * \text{LOG:YP(1)}$$

TAPL NOMDBE

EQ.TO TRADUCT :

\*\*\*\*\*

DBE:

$$\text{LOG:YP} = \text{C:1} + \text{C:2} * 2\text{AV:IKO} + .1 * \text{C:3} * \text{TFEND} * \text{TEMPS} + .9 * \text{LOG:YP(1)}$$

LEC FORM:

\*\*\*\*\*

LEC:

$$\text{LYP} = \text{C1} + \text{C2} * 2\text{AVIKO} + .1 * \text{C3} * \text{TFEND} * \text{TEMPS} + .9 * \text{LYP(1)}$$

EQ. IN APL FORM FOR NLEST :

\*\*\*\*\*

APL:

$$\text{YP[I]} \leftarrow \text{C[1]} + (\text{C[2]} * (\text{AV} + 2 \text{AV IKO})) + (.1 * \text{C[3]} * \text{TFEND[I]} * \text{TEMPS[I]}) + .9 * \text{YP[I-1]}$$

VV:

YP,IKO,TREND,TEMPS

VC:

C[1],C[2],C[3]

LMAX:

1



## TPL1 NOMDPE

EQ.TO TRANDUCT :

\*\*\*\*\*

DBE:

$$\text{LOG:YP} = \text{C:1} + \text{C:2} * 2\text{AV:IKO} + .1 * \text{C:3} * \text{TREND} * \text{TEMPS} + .9 * \text{LOG:YP(1)}$$

LEC FORM:

\*\*\*\*\*

LFC:

$$\text{LYP} = \text{C1} + \text{C2} * 2\text{AVIKO} + .1 * \text{C3} * \text{TREND} * \text{TEMPS} + .9 * \text{LYP(1)}$$

EQ. IN PL1 FORM FOR SIMUL :

\*\*\*\*\*

PL1:

$$\text{YP(I)} = \text{EXP}(\text{C(1)} + \text{C(2)} * (\text{AV} = \Delta\text{AV}(2, \text{IKO}) + .1 * \text{C(3)} * \text{TREND(I)} * \text{TEMPS(I)} + .9 * \text{LOG(YP(I-1))}))$$

VV:

YP, IKO, TREND, TEMPS

VC:

C(1), C(2), C(3)

LMAX:

1

## 3. TRADGP

Les fonctions reprises dans ce groupe ont pour objectif de réaliser la traduction d'une ou plusieurs équations DBE directement en APL et en PL1.

Le mode d'emploi de TRAD est le suivant :

TRAD 'LISTE'
--------------

où LISTE est une suite de noms d'équations DBE, séparés par une virgule  
réalise la traduction des différentes équations reprises dans LISTE en APL et PL1.

La fonction est à output implicite et imprime successivement pour chaque nom d'équation

- la forme DBE : (EQ.TO TRANDUCT : nom de l'équation)
- la forme LEC : (LEC FORM :)
- la forme APL : (EQ.IN APL FORM FOR NLEST :)
- VV : le vecteur contenant le nom des variables rencontrées au cours de la traduction, la première étant la variable endogène
- VC : le vecteur contenant la liste des coefficients rencontrés dans l'équation
- LMAX : le retard maximum rencontré lors de la traduction
- la forme PL1 : (EQ.IN PL1 FORM FOR SIMUL :)

Tous ces outputs, présentés sous la forme de vecteur alphanumérique, ont été personnalisés au cours de la traduction. C'est ainsi que pour retrouver la forme APL d'un nom d'équation appartenant à LISTE, il suffit de le faire précéder du littéral 'A'; il suffit de le faire précéder du littéral 'P' pour en retrouver le littéral PL1.

Pour retrouver le vecteur des noms de variables d'une équation, il suffit de le faire précéder du littéral 'V', celui des coefficients, du littéral 'C' ('C1' pour PL1) et celui du retard maximum, du littéral 'L'.

Il appartient à l'utilisateur de détruire ces variables si elles ne sont plus nécessaires.

TRAD 'NOM1,NOM2'

EQ.TO TRADUCT : NOM1

\*\*\*\*\*

DRE:

WP = C:1 + C:2 \* LOG:(YD - CP) + C:3 \* (LI - .9 \* LI(1))

LEC FORM:

\*\*\*\*\*

LEC:

WP = C1 + C2 \* L(YD - CP) + C3 \* (LI - .9 \* LI(1))

EQ. IN APL FORM FOR NLEST :

\*\*\*\*\*

APL:

WP[I]+C[1]+(C[2]\*YD[I]-CP[I])+C[3]\*LI[I]-.9\*LI[I-1]

VV:

WP,YD,CP,LI

VC:

C[1],C[2],C[3]

LMAX:

1

EQ. IN PL1 FORM FOR SIMUL :

\*\*\*\*\*

PL1:

WP(I)=C(1)+C(2)\*LOG(YD(I)-CP(I))+C(3)\*(LI(I)-.9\*LI(I-1))

EQ.TO TRADUCT : NOM2

\*\*\*\*\*

DPE:

DIF:MT0 = MT0(1) \* (C:1 \* (C:2 \* DIF:CP0 + C:3 \* DIF:IK0 + C:4 \* ST0 + C:5 \* DIF  
:XT0) + C:6 \* LOG:(PMT / PY))

LEC FORM:

\*\*\*\*\*

LEC:

DMT0 = MT0(1) \* (C1 \* (C2 \* DCP0 + C3 \* DIK0 + C4 \* ST0 + C5 \* DXT0) + C6 \* L(PM  
T ÷ PY))

EQ. IN APL FORM FOR NLEST :

\*\*\*\*\*

APL:

MT0[I]+MT0[I-1]+MT0[I-1]\*(C[1]\*(C[2]\*CP0[I]-CP0[I-1])+(C[3]\*IK0[I]-IK0[I-1])+(C[  
4]\*ST0[I])+C[5]\*XT0[I]-XT0[I-1])+C[6]\*PMT[I]÷PY[I]

VV:

MT0,CP0,IK0,ST0,XT0,PMT,PY

VC:

C[1],C[2],C[3],C[4],C[5],C[6]

LMAX:

1

EQ. IN PL1 FORM FOR SIMUL :

\*\*\*\*\*

PL1:

MT0(I)=MT0(I-1)+MT0(I-1)\*(C(1)\*(C(2)\*(CP0(I)-CP0(I-1))+C(3)\*(IK0(I)-IK0(I-1))+C(  
4)\*ST0(I)+C(5)\*(XT0(I)-XT0(I-1)))+C(6)\*LOG(PMT(I)/PY(I)))



ANOM1

$$WP[I] + C[1] + (C[2] \times \text{AYD}[I] - CP[I]) + C[3] \times LI[I] - .9 \times LI[I-1]$$

ANOM1

$$WP[I] + C[1] + (C[2] \times \text{AYD}[I] - CP[I]) + C[3] \times LI[I] - .9 \times LI[I-1]$$

PNOM1

$$WP(I) = C(1) + C(2) \times \text{LOG}(\text{YD}(I) - CP(I)) + C(3) \times (LI(I) - .9 \times LI(I-1))$$

VNOM2

MT0,CP0,IK0,ST0,XT0,PMT,PY

CNOM2

C[1],C[2],C[3],C[4],C[5],C[6]

LNOM2

1

C1NOM2

C(1),C(2),C(3),C(4),C(5),C(6)

## 4. Quelques caractéristiques

\100141:PLAN

016) 13.51.24 08/18/78 VILLE

. A P L . S V .

[WA

124700

\LOAD NAVYTRAD

SAVED 18.36.11 08/17/78

[WA

69828

)WIDTH 80

WAS 120

TRAD 'Z15'

EQ.TO TRANDUCT : Z15

\*\*\*\*\*

DPE:

$$DIF:MT0 = MT0(1) * (C:1 * (C:2 * DIF:CP0 + C:3 * DIF:IK0 + C:4 * ST0 + C:5 * DIF:XT0) + C:6 * LOG:(PMT / PY))$$

LFC FORM:

\*\*\*\*\*

LFC:

$$LMT0 = MT0(1) * (C1 * (C2 * LCP0 + C3 * LIK0 + C4 * LST0 + C5 * LXT0) + C6 * L(PMT / PY))$$

FO. IN APL FORM FOR NLFST :

\*\*\*\*\*

APL:

$$MT0[I] = MT0[I-1] + MT0[I-1] * (C[1] * (C[2] * CP0[I] - CP0[I-1]) + (C[3] * IK0[I] - IK0[I-1]) + (C[4] * ST0[I] + C[5] * XT0[I] - XT0[I-1]) + C[6] * LOG(PMT[I] / PY[I]))$$

VV:

MT0,CP0,IK0,ST0,XT0,PMT,PY

VC:

C[1],C[2],C[3],C[4],C[5],C[6]

LMAX:

1

EQ. IN PL1 FORM FOR SIMUL :

\*\*\*\*\*

PL1:

$$MT0(I) = MT0(I-1) + MT0(I-1) * (C(1) * (C(2) * (CP0(I) - CP0(I-1))) + C(3) * (IK0(I) - IK0(I-1))) + C(4) * ST0(I) + C(5) * (XT0(I) - XT0(I-1))) + C(6) * LOG(PMT(I) / PY(I))$$

)OFF

013 19.30.27 08/17/78 VIL

CONNECTED 0.02.03 TO DATE 382.03.29

CPU TIME 0.00.05 TO DATE 5.49.54

## 5. Introduction de nouvelles fonctions ou combinaisons de fonctions économétriques

Lors de la conception du langage économétrique condensé, nous avons défini un certain nombre d'opérateurs spécifiques à l'économètre, de même qu'un certain nombre de combinaisons de ces opérateurs qui peuvent porter sur une variable ou sur une expression.

L'utilisateur à la possibilité d'introduire de nouveaux opérateurs économétriques au fur et à mesure de ses besoins. Voici comment.

### A. Dans la fonction INITIAL

- Il faut introduire le libellé de la fonction dans FCT

NB : - dans la première partie de FCT, si ce libellé comprend 2 lettres

- dans la deuxième partie de FCT, si ce libellé comprend 1 lettre

- dans une nouvelle partie à créer devant les deux autres et séparée par un ou plusieurs blancs, si ce libellé comprend 3 lettres

- etc...

- Ayant introduit le libellé de la fonction dans FCT, il faut mettre à jour

STT : vecteur des indices précédant les débuts de noms de fonctions dans FCT

LGT : vecteur des longueurs du libellé des différentes fonctions

MASK : longueur du libellé le plus long

- Mettre dans NFCT, l'indice de la nouvelle fonction introduite dans FCT, si dans

n <u>nouvelle fonction</u> VAR (EXP)
---

n ne peut avoir de valeur. Cela permet d'envoyer si nécessaire un message d'erreur.



ex.: jusqu'ici 2 D VAR est la différence deuxième de VAR  
 mais 2 L VAR n'est pas admis. L'indice de L est 9 et se  
 trouve dans NFCT.

de même

mettre dans DFCT, l'indice de la nouvelle fonction intro-  
 duite dans FCT, si dans

n	<u>nouvelle fonction</u>	VAR
		(EXP)

n doit avoir 1 comme valeur par défaut.

- VDO reprend la liste des combinaisons de fonctions admises;  
 pour introduire une nouvelle fonction ou combinaison de fonc-  
 tion il faut en calculer la valeur de référence. Celle-ci se  
 calcule grâce à l'indice des fonctions dans le vecteur FCT.  
 NB : il faut tenir compte d'un indice pour l'ensemble des carac-  
 tères blancs séparant des fonctions de longueur différentes.  
 ex.: D est la 7e fonction dans FCT, elle a la valeur de réf-  
 érence 7  
R est la 8e fonction dans FCT, elle a la valeur de réf-  
 érence 8  
DR est composé de la 7e puis de la 8e fonction dans FCT,  
 elle a la valeur de référence 708  
 NB :  $7 \times 100 = 700$   
 $+ 8 = 708$   
 le facteur 100 permet d'avoir 99 fonctions élémen-  
 taires différentes, 9 était trop peu.

## B. Dans la fonction FT

Une fonction peut porter sur une variable ou sur une expression;  
 pour chacun de ces deux cas il faut donc introduire dans FT la  
 traduction souhaitée.

- Dans le cas où la nouvelle fonction ou combinaison de fonctions porte sur une variable, il faut insérer la traduction à la ligne
  - + EXIT
  - + l'indice de la valeur de référence concernée dans VDO

Pour composer la traduction,

CC : contient le nom de la variable sur laquelle porte la nouvelle fonction ou combinaison de fonction

LL : contient le retard portant sur la variable au départ de la traduction ('-', LG ou vide)

PR[2; FCF[I]] : contient la valeur du n de la dernière fonction

PR[2; FCF[I-1]] : contient la valeur du n de l'avant-dernière fonction

etc...

VL : doit être placé à l'endroit où l'indice de retard est le plus élevé; c'est cette valeur qui sera comparée au retard maximum déjà observé jusque là dans l'expression.

La ligne introduite doit avoir la forme suivante :

→ LM, OpV ← traduction, (TVL ← .....), ..

L'indicateur SWY est implicitement mis à 1.

- Dans le cas où la nouvelle fonction ou combinaison de fonctions porte sur une expression, il faut insérer la traduction à la ligne

+ EXP

+ l'indice de la valeur de référence concernée dans VDO

La ligne introduite doit avoir la forme suivante :

→ { LS1 , OpS1 ←        , OpS2 ←        , ....  
     LL11  
     LS12  
     LL22  
     autre

où S1 contiendra le signe impliqué pour la première fonction  
 S2 contiendra le signe impliqué pour la deuxième fonction  
 etc...

et où

LS1, LL11, ... sont des paragraphes représentant comment il  
 faut exécuter la traduction.

L'indicateur SWE doit être mis à 1.

Si la fonction introduite ne peut porter sur une expression, il  
 faut remplacer la traduction par  $\rightarrow$  EXP qui est une sortie de la  
 fonction avec message d'erreur.

C. Dans les fonctions DBIN et LECIN,

seuls les vecteurs  $\begin{cases} \text{TST} \\ \text{STTST} \\ \text{LGTST} \end{cases}$  et  $\begin{cases} \text{TRAD} \\ \text{STTRAD} \\ \text{LGTRAD} \end{cases}$  doivent être mis à jour

Nous avons ainsi décrit les modifications qu'il faut porter au  
 traducteur APL. Les modifications équivalentes doivent évidemment  
 encore être apportées au traducteur PL1. Les fonctions concernées  
 sont INITIAL1 (voir INITIAL) et FT1 (voir FT) du groupe TPL1GP;  
 LECIN et DBIN sont en effet communes aux trois groupes TAPLGP,  
 TPL1GP et TRADGP.



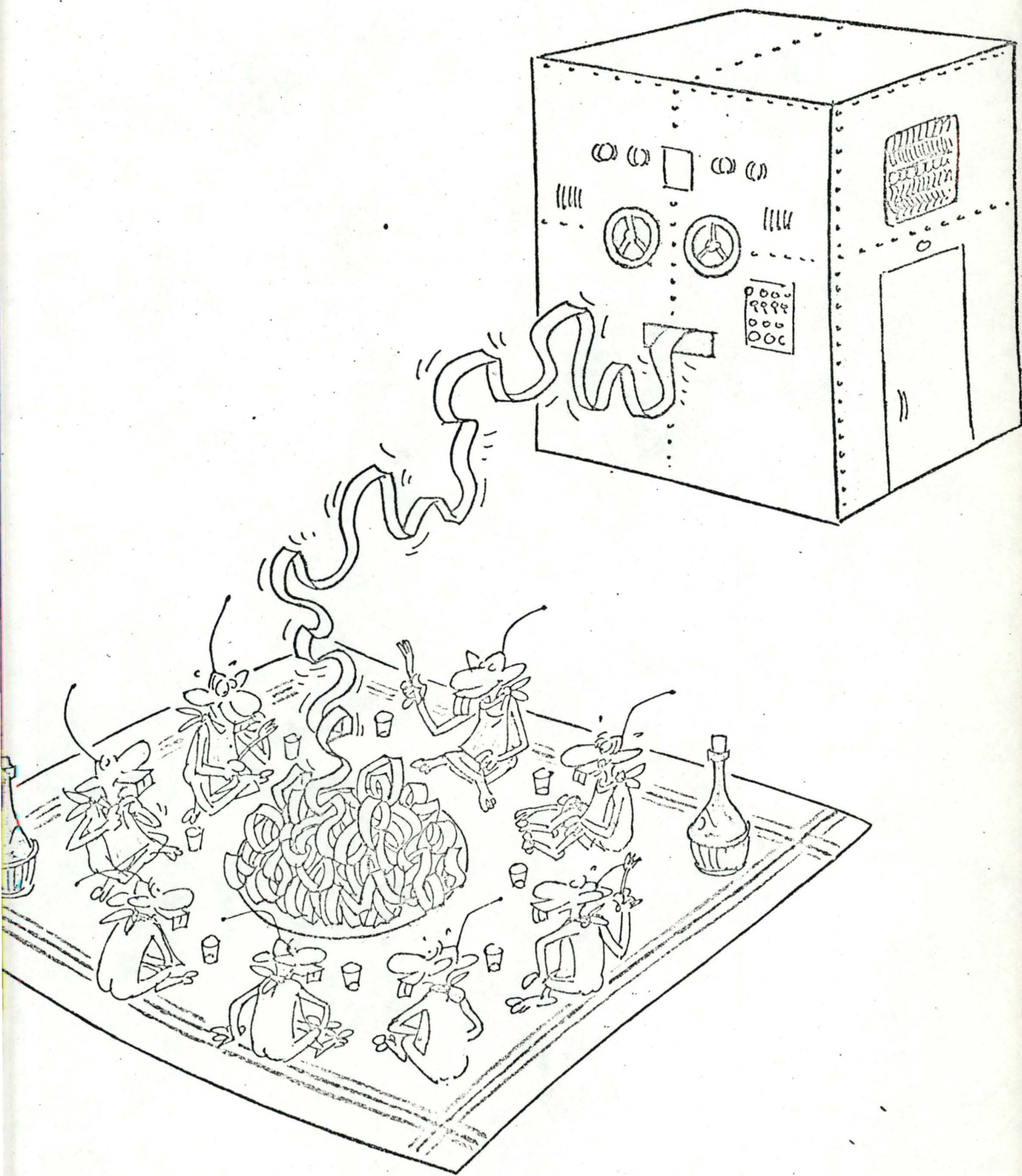
DOSSIER DESCRIPTIF DES FONCTIONS  
COMPOSANT LE SOFTWARE DE TRADUCTION  
D'UNE EQUATION ECONOMETRIQUE EN SYNTAXES  
APL ET PL1.

1. TAPLGP
2. TPL1GP
3. TRADGP

DD. 1

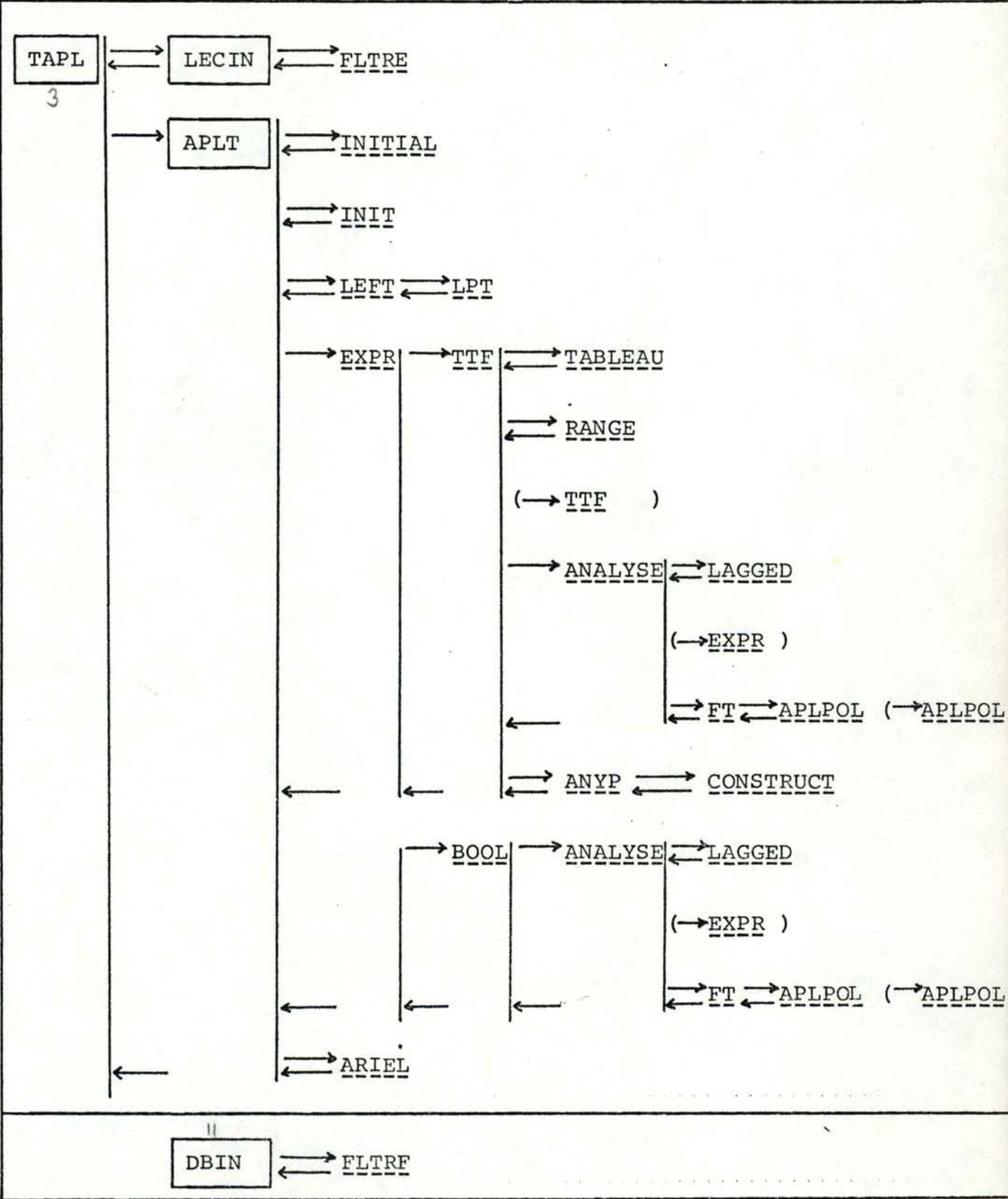
DD.75

DD.76



1. TAPLGP

L'enchaînement des différentes fonctions du groupe TAPLGP se schématise comme suit :





où -  $\longrightarrow$  représente l'appel d'une fonction

-  $\longleftarrow$  représente le retour à la fonction appelante

-  $(\rightarrow)$  représente un appel récursif

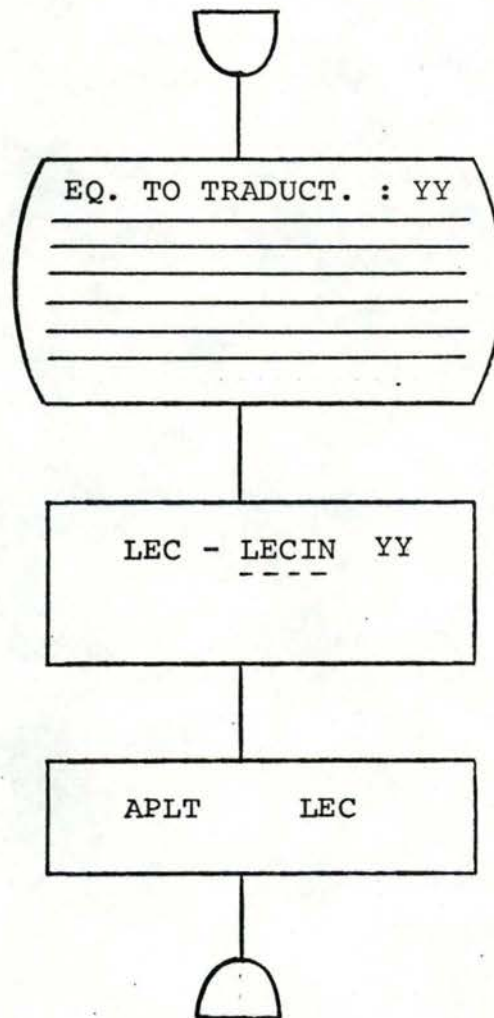
-  représente une fonction directement utilisable par l'utilisateur

Fonction





TAPL YY

\* Objet : transformer une équation écrite en DBE en une équation écrite en syntaxe APL.

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.		
							YY
			<input type="checkbox"/> IO				
						YY	
<u>LECIN</u>	YY	—	(cf. <u>LECIN</u> )	—	LEC	—	
APLT	LEC	—	(cf. APLT)	—	—	ZZ W VC IMAX	



\* Variables utilisées

- ☐ IO : Index origine (☐ IO = 1)
- YY : nom interne à TAPL de l'équation à traduire
- LEC : YY mis sous forme Langage Econométrique Condensé
- ZZ : contient le littéral de l'équation traduite en syntaxe APL
- VV : contient la liste des variables utilisées dans l'équation
- VC : contient la liste des coefficients utilisés dans l'équation
- LMAX : retard maximum rencontré dans l'équation

Fonctions appelées

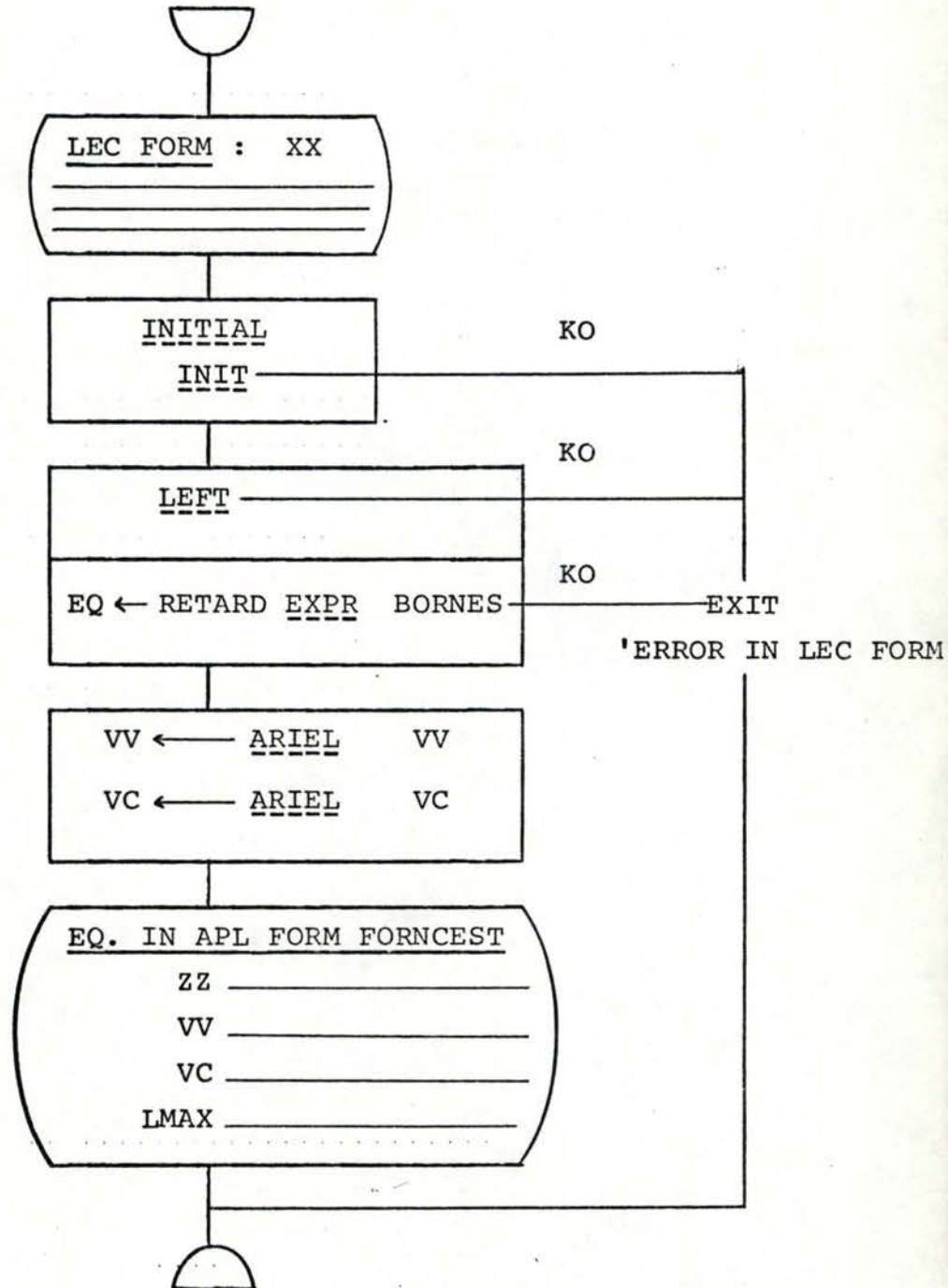
- LECIN : transforme une équation DBE dans une syntaxe LEC
- APLT : transforme une équation LEC dans une syntaxe APL.

Fonction


APLT . XX

\* Objet : transforme une équation LEC dans une syntaxe APL -  
APLT est la routine principale qui ordonne les principales étapes de la traduction.

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.		
							XX
			<input type="checkbox"/> IO				
						XX	
<u>INITIAL</u>			(cf. <u>INITIAL</u> )				
			LTR LTRF NUM CAR				
<u>INIT</u>		XX LTR, LTRF NUM CAR	(cf. <u>INIT</u> )	NP	ERR		
			BORNES				
<u>LEFT</u>		XX LTR NUM BORNES	(cf. <u>LEFT</u> )	ZZ, VV, BORNES	ERR		
			VC, LMAX OPA, OP, OPB OPR, OPRB RETARD SW Y S				
<u>EXPR</u>	BORNES	XX cf. <u>INITIAL</u> LTR, LTRF, NUM, VC, LMAX OPA, OP, OPB OPR, OPRB RETARD SW Y NP	(cf. <u>EXPR</u> )	ZZ, VV, VC	EQ		



<u>ARIEL</u>	WV		(cf. <u>ARIEL</u> )		WV		
<u>ARIEL</u>	VC				VC		
						ZZ WV VC LMAX	
			P,SA				

\* Variables utilisées

☐ IO : Indix origine ( ☐ IO = 1 )  
 XX : nom interne à APLT de l'équation à traduire  
 LTR : ensemble des lettres  
 LTRF : ensemble des lettres soulignées  
 NUM : ensemble des chiffres plus le point  
 CAR : ensemble des caractères admis dans LEC  
 NP : vecteur contenant le niveau de parenthétisation de l'équation à traduire  
 ERR : indicateur d'erreur  
 BORNES : pointeurs courant sur l'équation  
 ZZ :  
 VV : } cf. TAPL  
 VC : }  
 LMAX : }  
 OPA : opérateurs arithmétiques (LEC)  
 OP : opérateurs arithmétiques (APL)  
 OPB : opérateurs booléens (LEC)  
 OP1,  
 STOP1,  
 LGOP1 : opérateurs booléens (APL)  
 OPR : vecteur équivalent à XX contenant une représentation numérique des opérateurs  
 OPRB : vecteur arithmétiques contenant une représentation numérique des opérateurs booléens  
 RETARD : retard portant à un moment donné sur une expression  
 SW : switch indiquant s'il faut ou non introduire des parenthèses en APL  
 Y : vecteur contenant la traduction d'un élément de base  
 S : switch indiquant s'il faut entrer dans l'analyse d'une expression arithmétique ou booléens  
 EQ : vecteur contenant la traduction d'une expression  
 P, SA : variables de travail utilisées par TTF

Fonctions appelées

- INITIAL : initialise les différentes fonctions traductibles par APLT
- INIT : supprime les blancs, teste les caractères admis et la parité des ().
- LEFT : analyse puis traduit la partie gauche de l'équation (avant le signe =)
- EXPR : fonction récursive d'analyse d'une expression (au départ, c'est toute la partie droite de l'équation qui est considérée)
- ARIEL : VV et VC enregistrent toutes les variables rencontrées au cours de la traduction; certaines sont rencontrées un nombre quelconque de fois. ARIEL nettoie VV et VC des éléments en double.

NB : \_\_\_\_\_ = \_\_\_\_\_  $\equiv$  Equation

$\longleftrightarrow$  =  $\longleftrightarrow$   
 partie gauche = (      partie droite      )  $\equiv$  expression

LEFTEXPR

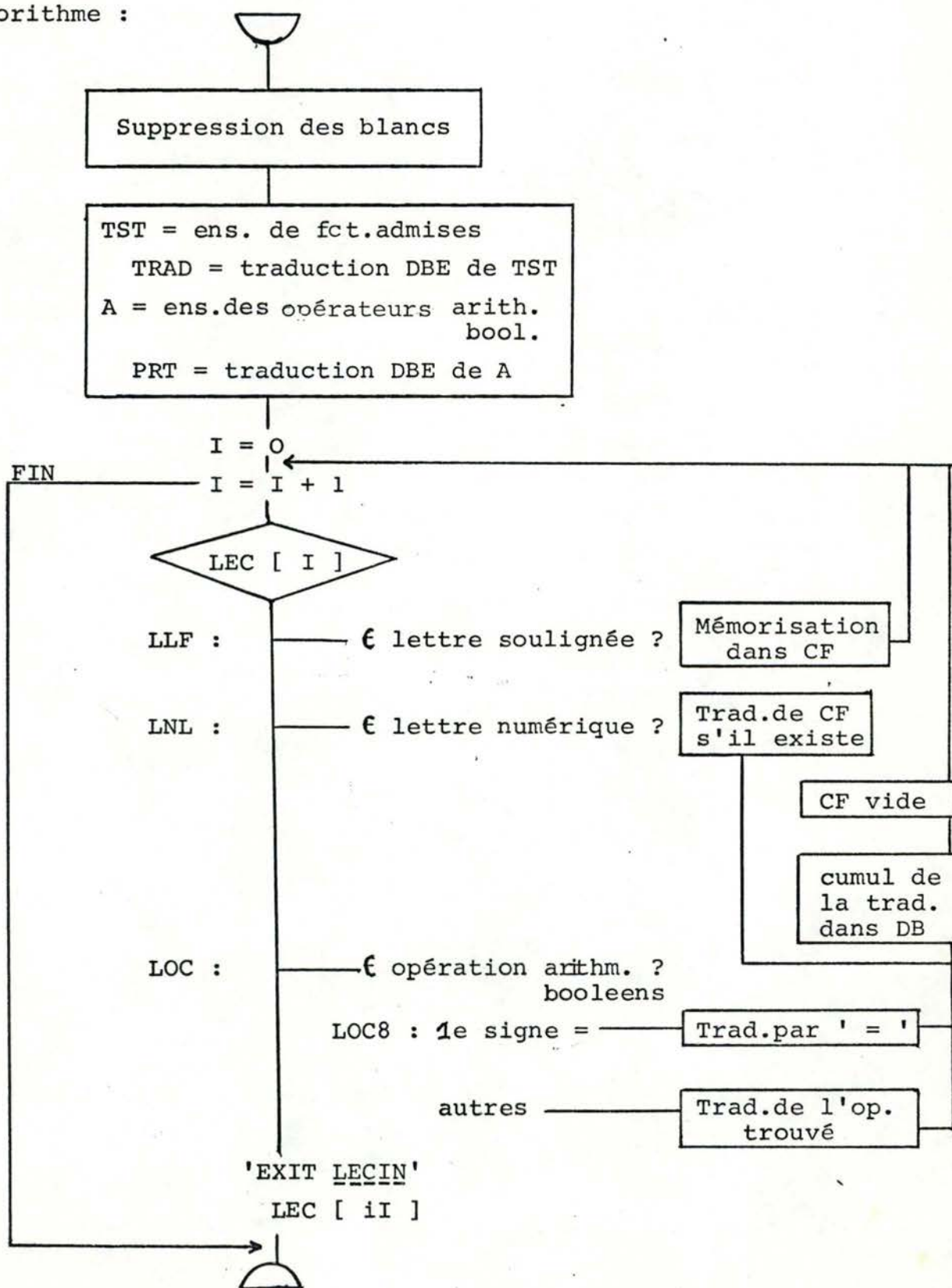


Fonction

DB ← DBIN LEC

\* Objet : traduit une équation de LEC en lisible pour DBE

\* Algorithme :



\*

Fonctions	variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>	—
							LEC
							DB
			<input type="checkbox"/> IO				
			TST, STST, LGIST  TRAD, STTRAD, LGTRAD  A  PRT, STPRT, LGPRT				
			I  CF  J, T, B				
<u>FLTRF</u>	CF	J	(cf. FLTRF)		V		
					DB		

\* Variables utilisées

☐ IO : définit l'origine des vecteurs utilisés ( ☐ IO = 1)  
 LEC : équation en LEC à traduire  
 DB : résultat de la transposition de LEC en DBE  
 TST,  
 STTST,  
 LGTST : ensemble des fonctions admises : TST[STTST[x]+ LLGTST[x]]  
 TRAD,  
 STTRAD,  
 LGTRAD : traduction correspondante à TST  
 A : ensemble des opérateurs arithmétiques, booléens admis  
 PRT, STPRT,  
 LGPRT : traduction correspondante à A  
 I : indice courant sur LEC  
 CF : vecteur mémorisant les fonctions trouvées  
 J : indice courant sur TST et TRAD dans FLTRF  
 B : permet de savoir quel élément de A on a rencontré  
 T, B=8 : permettent de savoir si le signe = rencontré est le  
 premier ou non; (T = 0 ou 1)  
 si oui, il s'agit du premier signe d'affectation,  
 si non, d'un signe d'égalité booléenne

Fonctions appelées

FLTRF : recherche dans TST la combinaison de fonctions présente dans CF,  
 et on renvoie la traduction dans V.

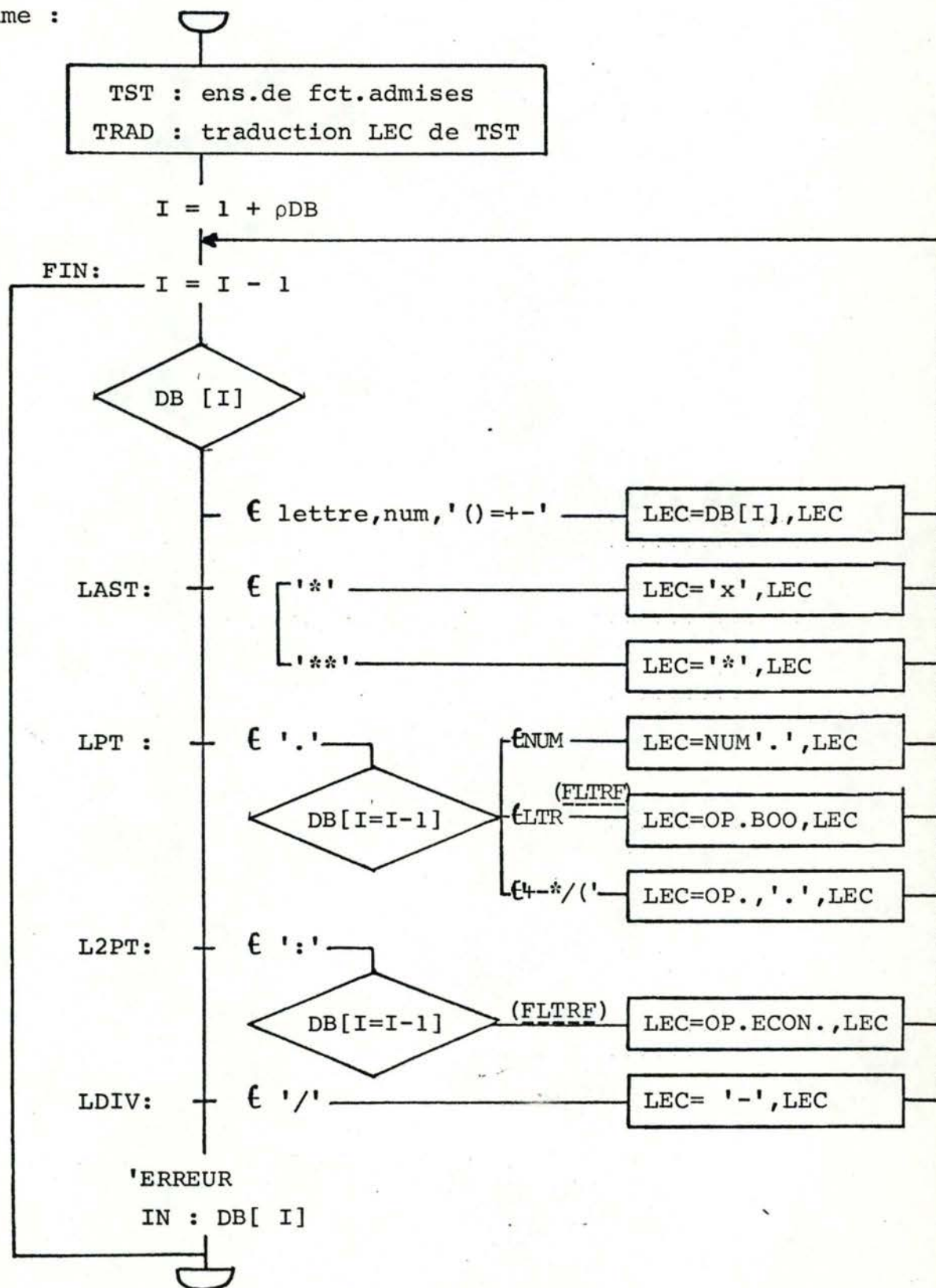


Fonction

LEC  $\leftarrow$  LECIN DB

\* Objet : traduit une équation DBE en LEC

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>	
							DB LEC
			<input type="checkbox"/> IO				
			TST,STTST, LGIST TRAD,STTRAD, LGTRAD NUM,LTR				
			I, J A, S, AST M, V				
<u>FLTRF</u>	M	J	(cf. <u>FLTRF</u> )		V		
					LEC		

\* Variables utilisées

☐ IO : index origine  
 DB : équation en DBE à traduire  
 LEC : résultat de la transposition de DB en LEC  
 TST, STTST, : ensemble des fonctions admises : TST[ STTST[ x] +  
 LGTST : 1LGTST[ x] ]  
 TRAD, STTST, : traductions correspondantes à TST  
 LGTST :  
 I : indice courant sur DB  
 J : indice courant sur TST et TRAD dans FLTRF  
 M : vecteur mémorisant les fonctions économétriques ou  
 opérateurs booléens rencontrés dans DB  
 V : contient la traduction LEC de M  
 A, S, AST : variables intermédiaires

Fonction appelée

FLTRF : recherche dans TST, la combinaison de fonctions présente dans M, et en renvoie la traduction dans V

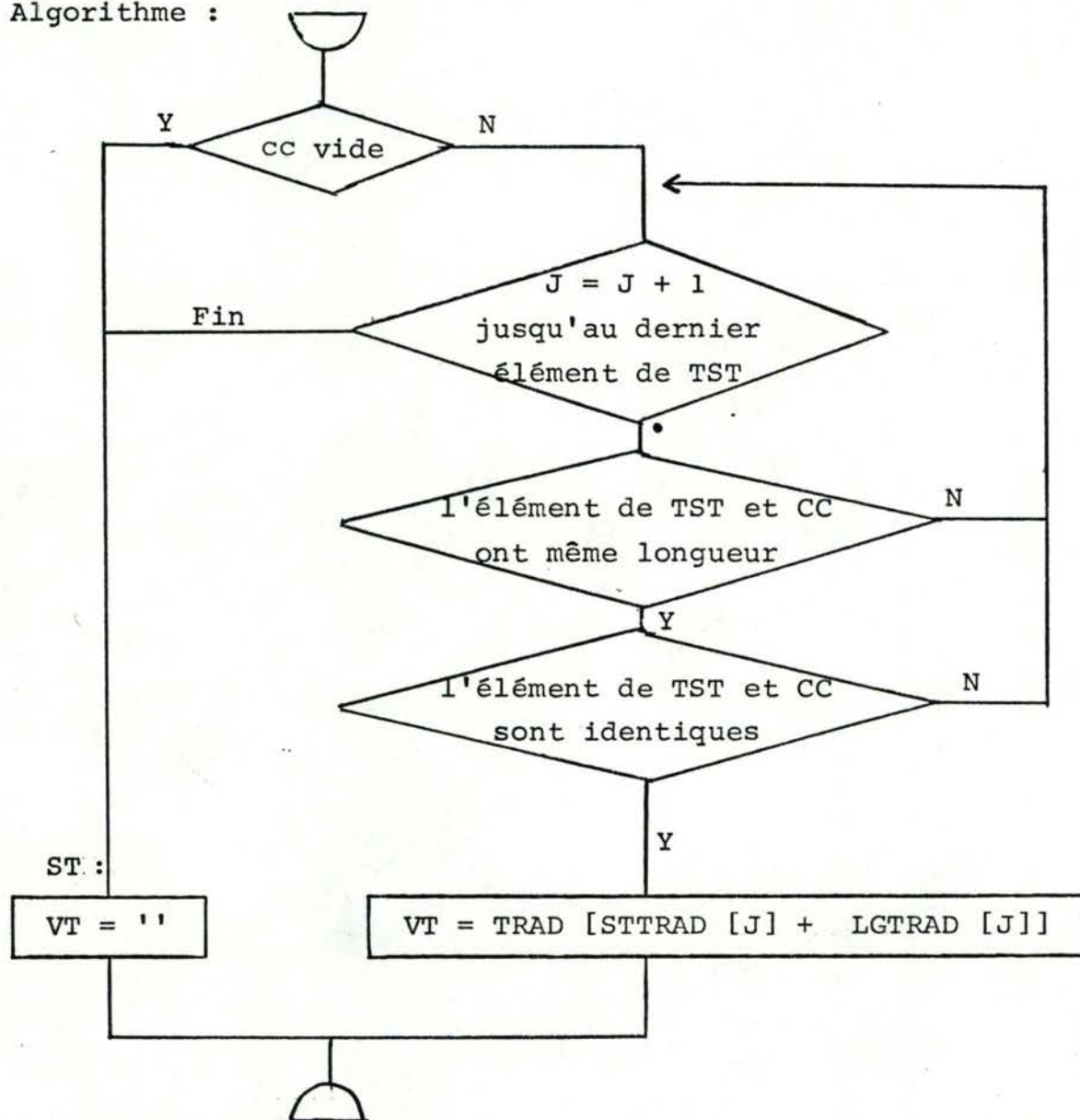


Fonction

$$VT \leftarrow \underline{FLTRF} \ CC$$

\* Objet : recherche dans TST la combinaison de fonctions présente dans CF, et en renvoie la traduction dans VT.

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>	
							CC VT
		<input type="checkbox"/> IO					
	CC						
		TST STTST LGTST J TRAD STTRAD LGTRAD					
					VT		

\* Variables utilisées

CC : contient la fonction ou combinaison de fonction à traduire

VT : contiendra la traduction de la fonction ou combinaison de fonction trouvée dans CC

TST,  
STTST,  
LGTST : ensemble des fonctions admises  $TST [STTST [x] + LGTST[x]]$

TRAD,  
STTRAD,  
LGTRAD : traduction correspondante à TST

J : indice courant sur TST et TRAD.



Fonction

INITIAL

---

\* Objet : initialise les différentes fonctions que l'on peut rencontrer dans l'équation  
et  
définit également les différentes combinaisons de ces  
fonctions que l'on peut rencontrer

\* Algorithme : —

\* Variables utilisées

FCT, STT, LST : fonctions que l'on peut rencontrer dans une équation

PARAM1, STP1,  
LSP1  
PARAM2, STP2,  
LSP2

: traduction des fonctions de FCT en APL

MASK : masque dont la valeur est la longueur maximale d'un nom de fonction de FCT  
ex.: AV pour average est de longueur 2  
L pour logarithme est de longueur 1

VDT : vecteur de valeurs permettant de faire une table de décision pour les éléments de base ceux-ci sont :

VDT

0 un coefficient C  
2 un numérique N  
1 une variable V  
4 une variable avec retard V(  
8 une expression E  
24 une expression avec retard E(

VDO : vecteur de valeurs permettant de faire une table de décision pour les combinaisons de fonctions rencontrées en partie droite les combinaisons admises sont :

VDO

0 une variable  
7 D différence première d'une variable  
8 R Rapport d'une variable  
9 L logarithme d'une variable  
10 A Absolu d'une variable  
708 DR  
807 RD  
907 LD  
908 LR  
709 DL  
809 RL

- 1 AV moyenne d'une variable
- 2 AG taux de croissance moyen d'une variable
- 3 MX maximum d'une variable
- 4 MN minimum d'une variable

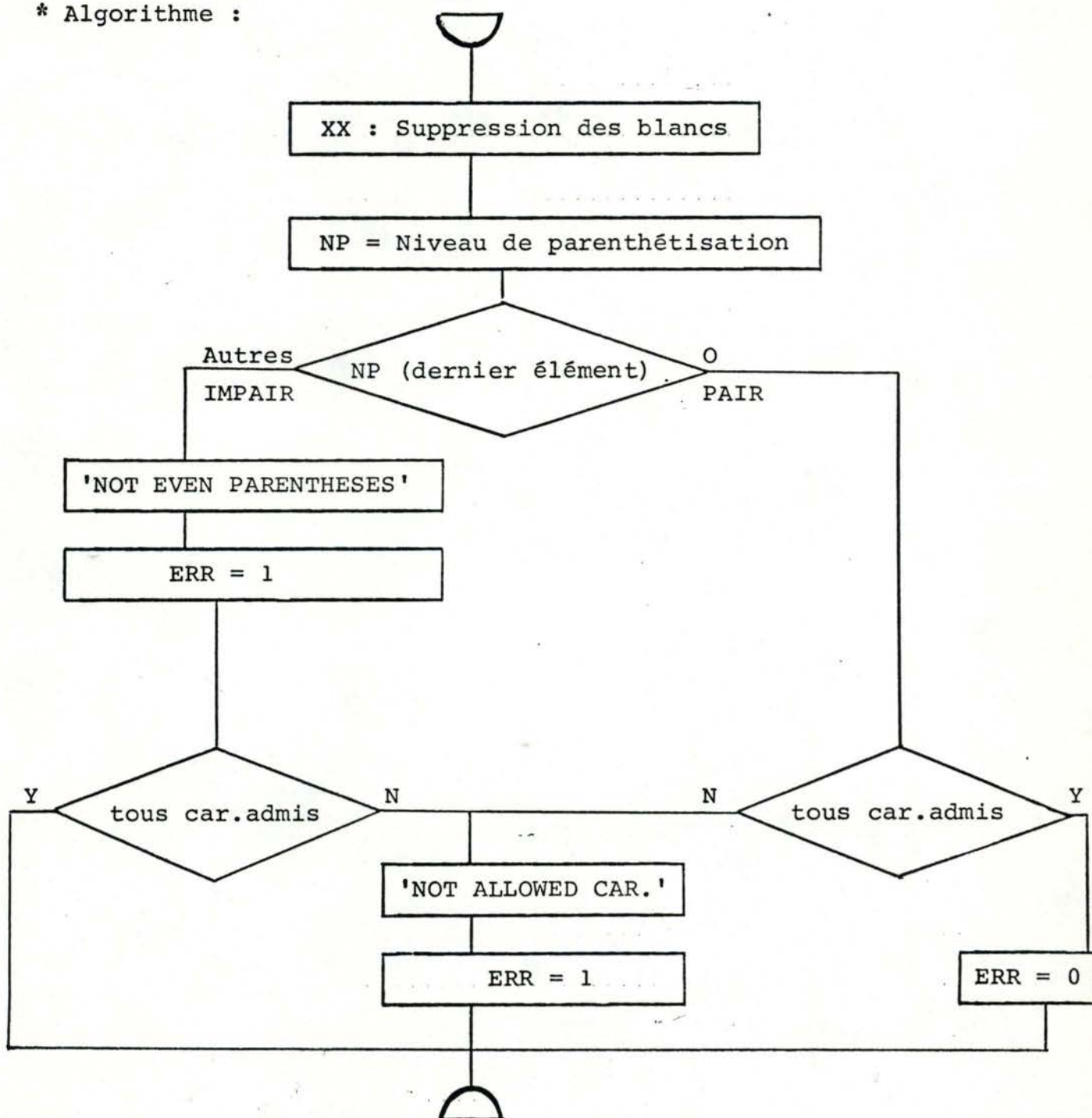


Fonction

Ehh ← INIT

\* Objet : supprime les blancs, teste les caractères admis et la parité des parenthèses

\* Algorithme :



\*

fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>	
		XX					
		LTR, LTRF, NUM, CAR					
				NP			
			A				
					ERR		

\* Variables utilisées

XX : nom de l'équation à traduire

NP : niveau de parenthétisation de XX

A : vecteur de présence de caractère inadmis

ERR : indicateur d'erreur

LTR	:	}	cf. APLT
LTRF	:		
NUM	:		
CAR	:		



Fonctions

ERR ← LEFT

et

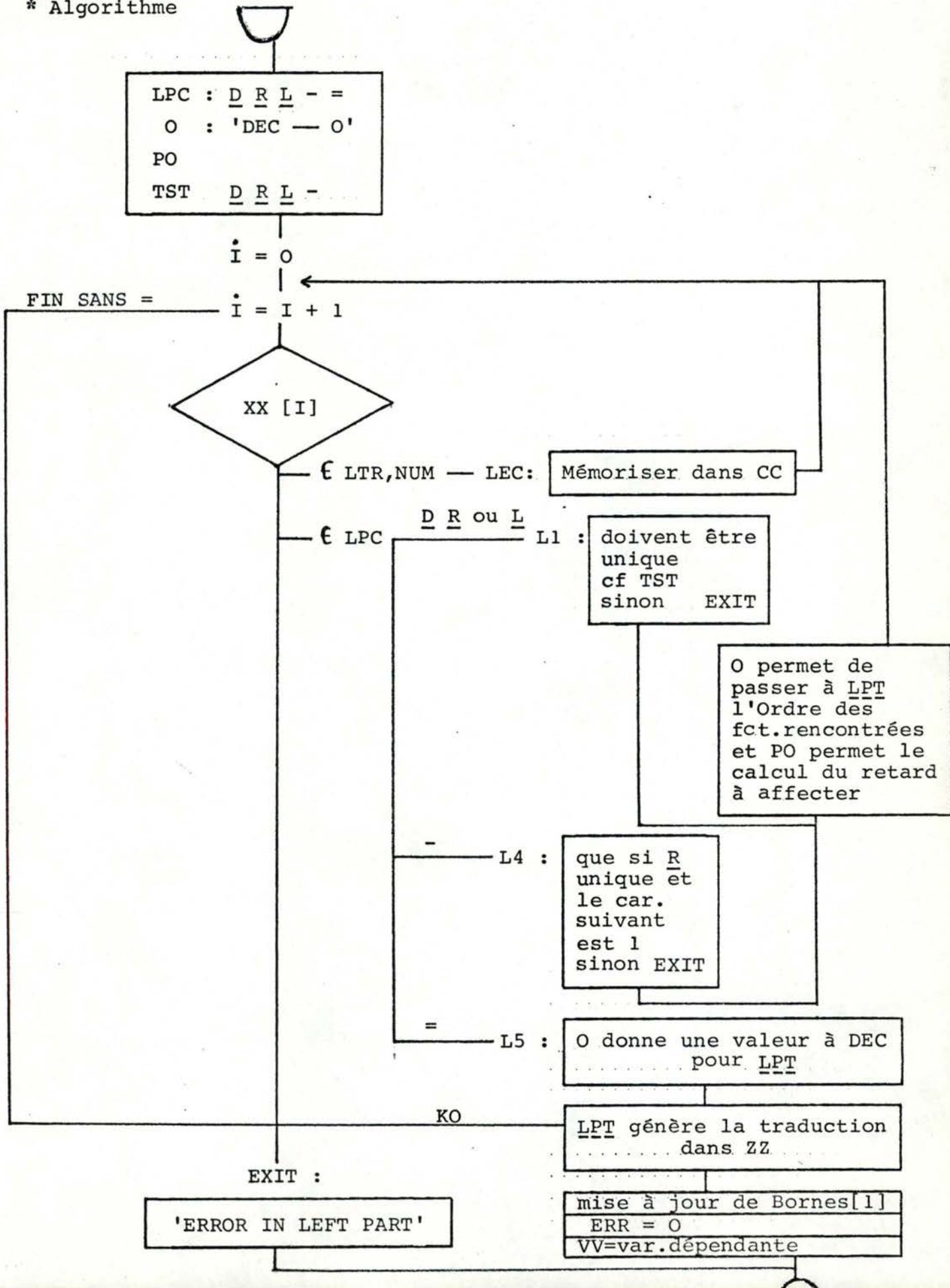
ERL ← LPT

- \* Objet : LEFT analyse la partie gauche de l'équation (avant le signe =) puis  
LPT génère la traduction APL de la partie gauche en fonction d'une valeur de décision reçue de LEFT

Rem.: Il n'y a qu'un nombre limité de fonctions qui sont acceptées en partie gauche :

VAR =  
D VAR =  
R VAR =  
L VAR =  
R VAR-1 =  
D L VAR =  
D R VAR =  
D R VAR-1 =  
L D VAR =  
L R VAR =  
L R VAR-1 =  
R D VAR =  
R L VAR =  
R D VAR-1 =  
R L VAR-1 =

## \* Algorithme



\*

Fonctions	Variables					Param.
	INPUT		LOCALES	OUTPUT		FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	
		<input type="checkbox"/> IO			<input type="checkbox"/>	
		XX				
		LTR,NUM, BORNES				
			i LPC,TST O (DEC) PO CC A X			
<u>LPT</u>		<input type="checkbox"/> IO ZZ DEC PO CC	A X Y 1,Y2,Y 3		ERI	
				ZZ VV BORNES	ERR	



\* Variables utilisées

☐ IO : définit l'origine des vecteurs ( ☐ IO = 1 )  
 XX : nom de l'équation à traduire  
 LTR : } cf APLT  
 NUM : }  
 BORNES : pointeurs courant sur l'expression  
 I : indice courant sur XX  
 LPC : fonctions admises en partie gauche DRL - et signe =  
 TST : TEST de présence de ces fonctions (contraintes de présence unique d'une fonction)  
 O, (DEC) : 'DEC ← 0' : Vecteur mémorisant l'ordre des fonctions rencontrées  $\begin{Bmatrix} \underline{D} & \underline{R} & \underline{L} & - \\ 1 & 2 & 3 & 4 \end{Bmatrix}$   
 ex.: DEC = 21  $\Rightarrow$  R D VAR  
 DEC = 124  $\Rightarrow$  D R VAR-1  
 DEC = 0  $\Rightarrow$  VAR  
 PO : Vecteur permettant de calculer le retard à affecter  
 ex.: PO = 0 0 par ex.: VAR  
 PO = 1 0 par ex.: D VAR  
 PO = 0 1 par ex.: L R VAR  
 PO = 1 1 par ex.: D R VAR  
D R VAR-1  
 CC : mémorise le littéral de la variable rencontrée  
 A, X : variables intermédiaires  
 Y1, Y2, Y3: variables intermédiaires  
 ZZ : }  
 VV : } cf. APLT  
 ERR, ER1: indicateurs d'erreur

Fonctions appelées :

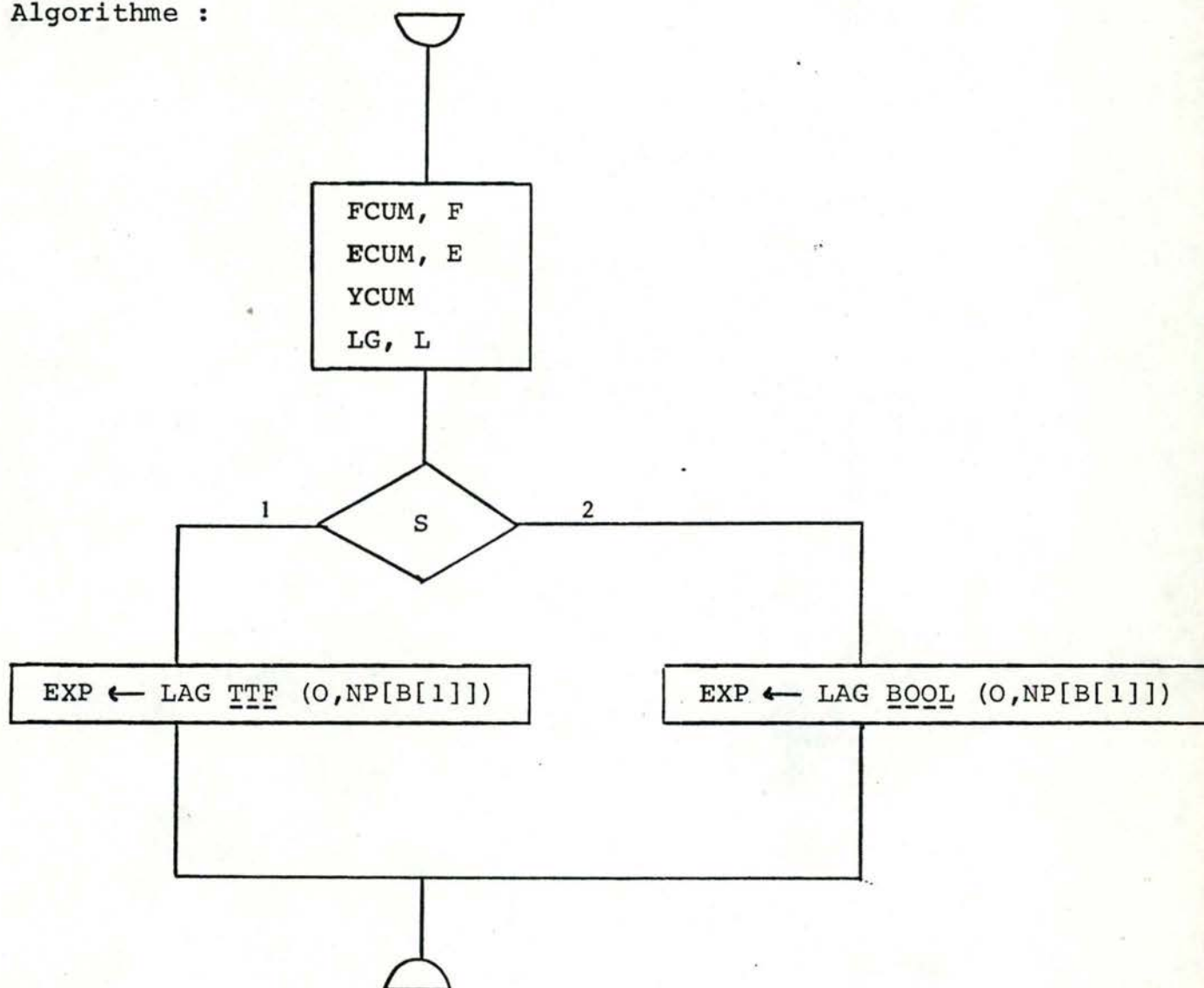
LPT : génère la traduction APL de la partie gauche en fonction d'une valeur de décision (DEC) reçue de LEFT

Fonction


EXP  $\leftarrow$  LAG EXP R B

\* Objet : fonction réursive d'analyse d'une expression

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.		
							B
							LAG
							EXP
		<input type="checkbox"/> IO					
		XX cf. INITIAL LTR, LTRF, NUM VC, IMAX OPA, OP, OPB, OPR, OPRB SW Y NP					
			FCUM, F ECUM, E YCUM FONCTION LG, L				
<u>TTF</u>	LAG N=(O, NP[B[1]])	-	(cf. <u>TTF</u> )	N B		-	
					EXP		
<u>BOOL</u>	LAG N=(O, NP[B[1]])	-	(cf. <u>BOOL</u> )	N B		-	
					EXP		



# \* Variables utilisées

<input type="checkbox"/> IO	: définit l'origine des vecteurs utilisés ( <input type="checkbox"/> IO = 1)
B	: Bornes : pointeurs vers le début et la fin de l'expression considérée
LAG	: retard portant sur l'expression considérée
EXP	: traduction APL de l'expression considérée
XX	: nom de l'équation à traduire
cf. <u>INITIAL</u>	: cf. <u>INITIAL</u>
LTR, LTRF, NUM	: } cf. APLT
VC, LMAX	
OPA, OP, OPB	
OPR, OPRB	
SW	
Y	
NP	: }
F	: vecteur contenant la traduction d'un terme
FCUM	: vecteur cumulant la traduction de termes successifs
E	: vecteur contenant la traduction d'un facteur
ECUM	: vecteur cumulant la traduction de facteurs successifs
LG, L	: variables servant à mémoriser ou cumuler les retards portant sur les différentes variables
N=(0, NP[B[1]]): vecteur représentant les niveaux auxquels on se trouve	
$N[1] = \begin{cases} 0 \\ 1 \\ 2 \end{cases} = \begin{cases} \text{niveau d'un Terme} \\ \text{niveau d'un Facteur} \\ \text{niveau d'un Exposant} \end{cases}$	
N[2] = NP[B[1]] = niveau de parenthétisation auquel on se trouve, c'est-à-dire celui du début de l'expression considérée (B[1] = pointeur vers le début de l'expression considérée).	

## Fonctions appelées :

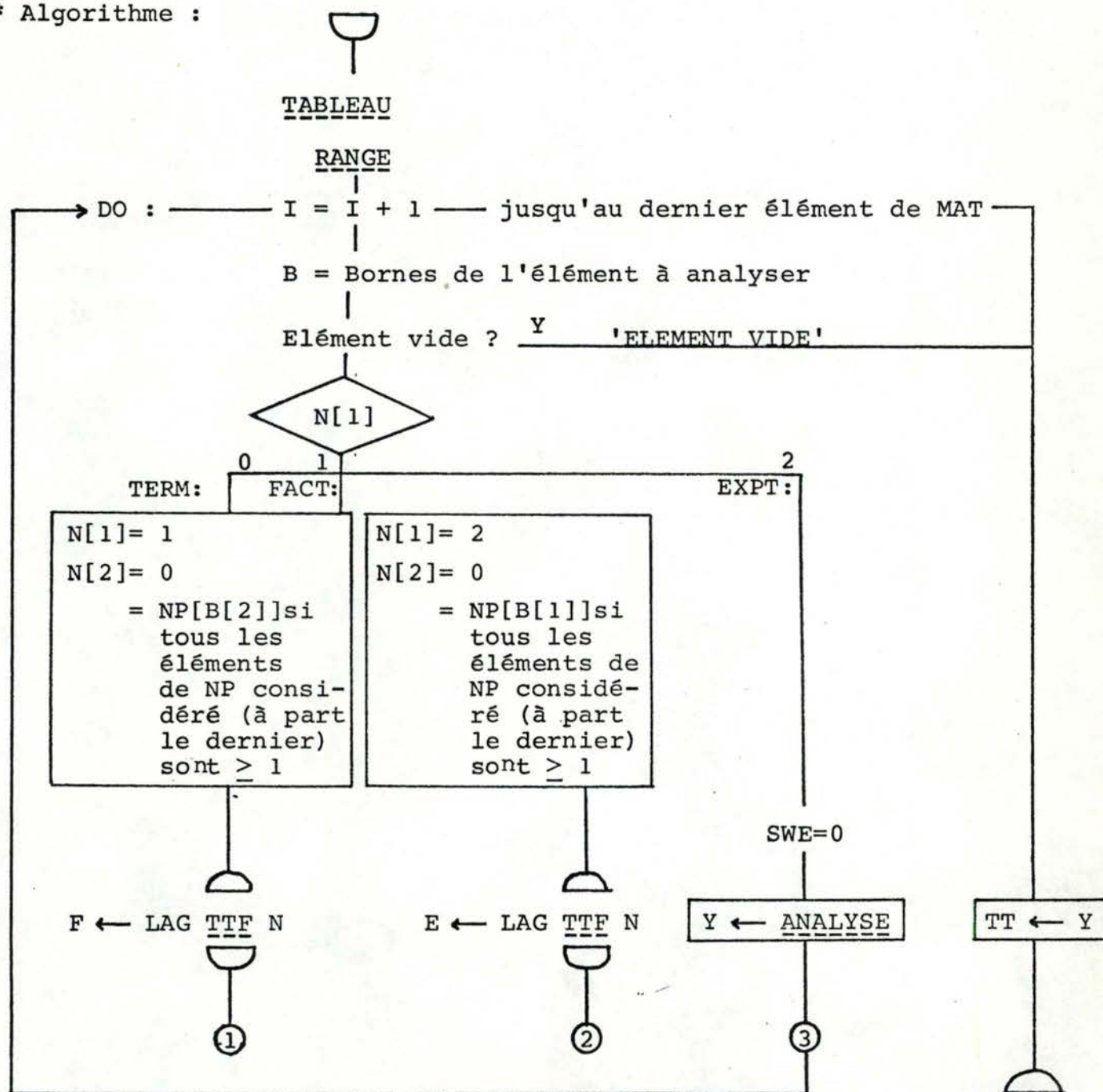
- TTF : routine appelée par EXPR pour décomposer une expression arithmétique en Termes, Facteurs, et Exposants, puis en rencontrer la traduction.
- BOOL : routine appelée par EXPR pour analyser puis traduire une expression booléenne.

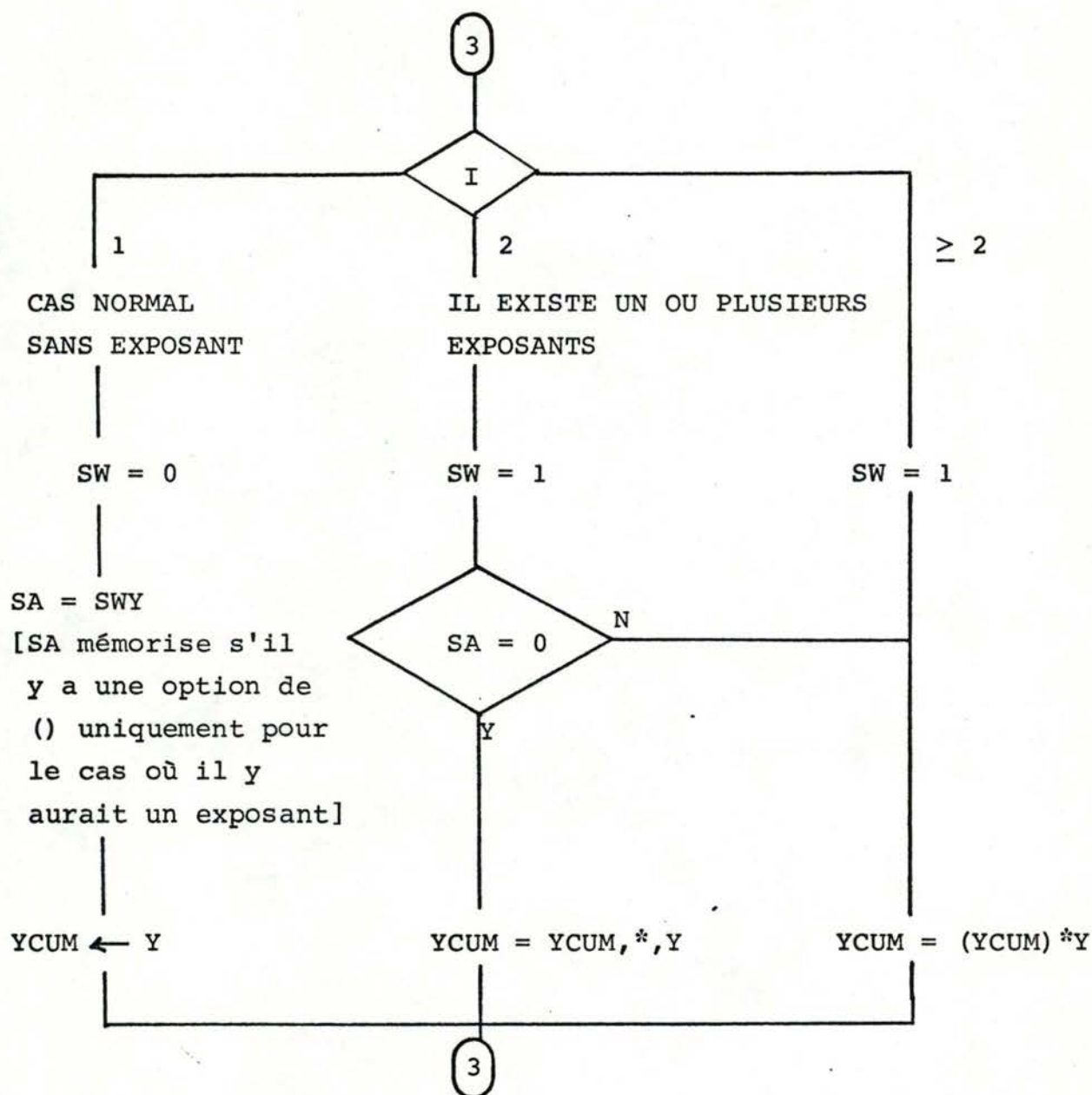
Fonction

$$TT \leftarrow LAG \underline{TTF} N$$

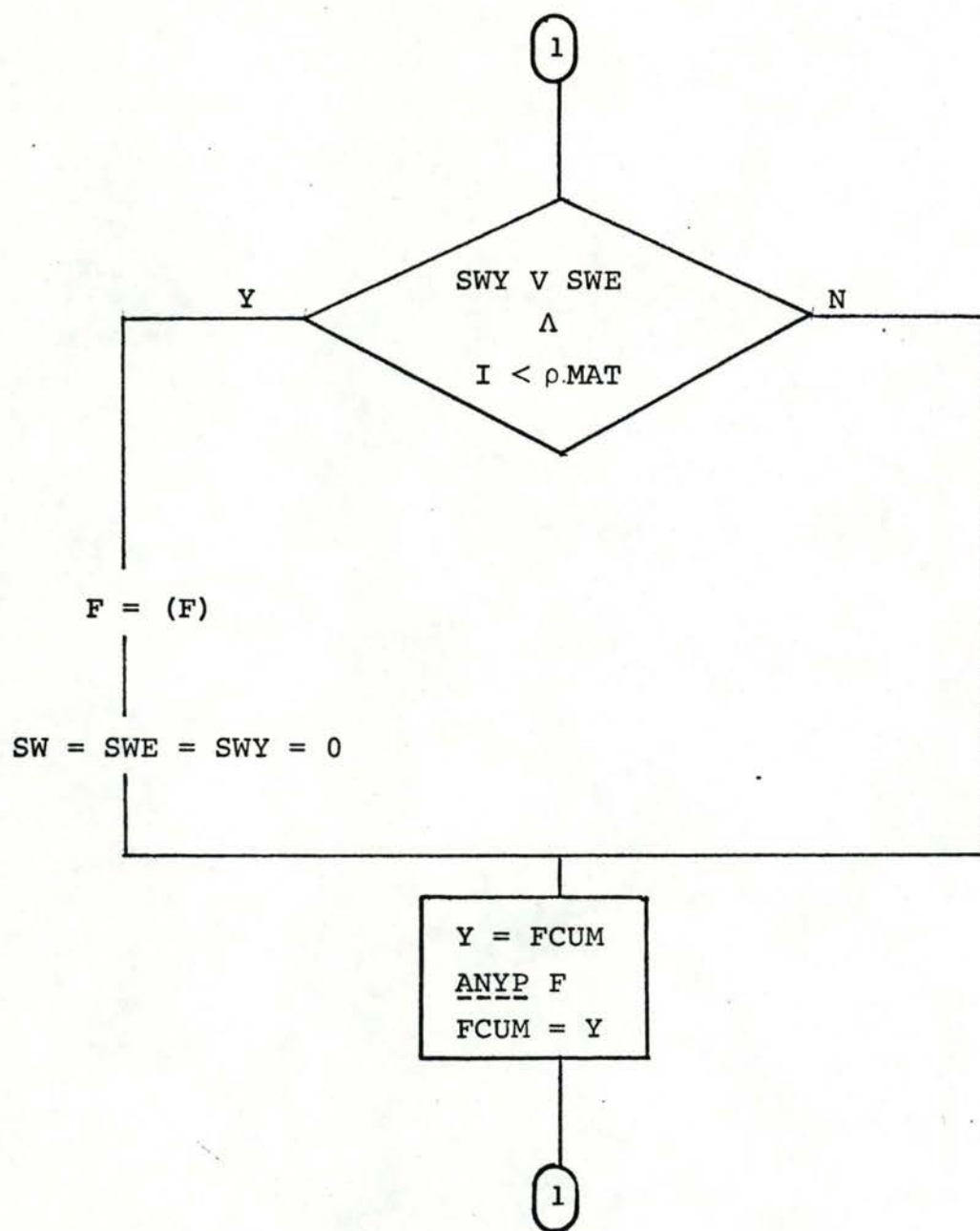
\* Objet : Analyse récursivement la structure des opérateurs binaires; après traduction de chaque élément de base, elle recompose l'expression de départ

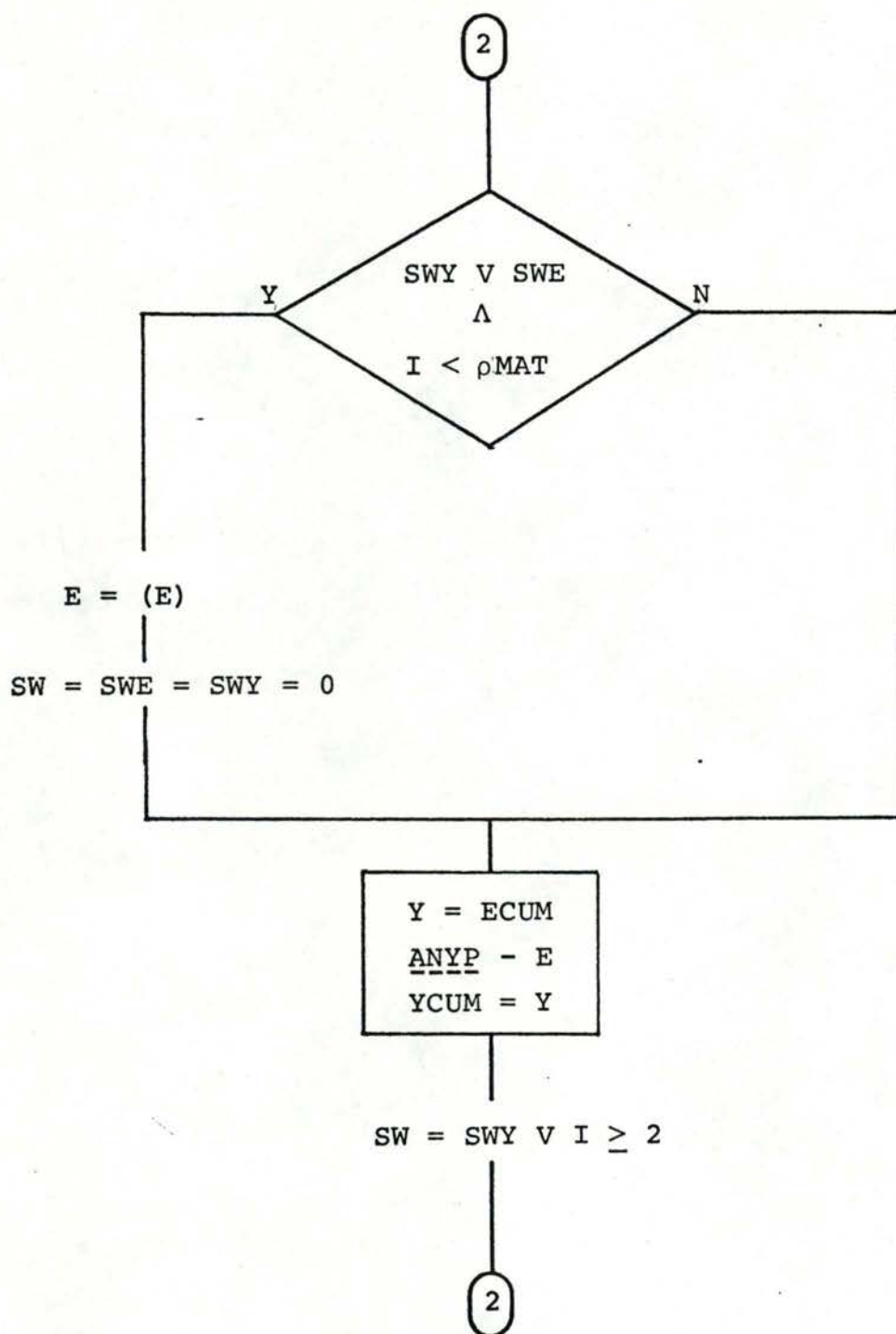
\* Algorithme :











\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>	
							N LAG TT
		<input type="checkbox"/> IO					
			MAT I, SA				
		P					
		B					
<u>TABLEAU</u>			cf. <u>TABLEAU</u>				
<u>RANGE</u>			cf. <u>RANGE</u>				
		XX		B			
		Y, YCUM E, ECUM F, FCUM SWY, SWE, SW					
		NP		P			
<u>TTF</u>	0 N= 1 P 2 LAG		cf. <u>TTF</u>		E ou F		
<u>ANYP</u>	E ou F		cf. <u>ANYP</u>	E ou F			
				Y, YCUM E, ECUM F, FCUM SWY, SWE SW			
<u>ANALYSE</u>			cf. <u>ANALYSE</u>		Y		
					TT		



\* Variables utilisées

N	: voir <u>EXPR</u>
LAG	: retard portant sur l'expression à analyser
TT	: résultat de la traduction
<input type="checkbox"/> IO	: définit l'origine des vecteurs
MAT	: tableau de termes, facteurs ou exposants (voir <u>TABLEAU</u> )
I	: indice courant sur MAT
P, SA	: variable intermédiaire
B	: Bornes de l'expression à considérer
XX	: nom propre à APLT de l'équation à traduire
NP	: vecteur contenant le niveau de parenthétisation de XX
Y, YCUM E, ECUM F, FCUM	{ : voir <u>EXPR</u>
SWY, SWE, SW	
	: switch permettant de définir s'il faut ou non introduire des ()
- SWY	{ : voir <u>FT</u>
- SWE	
- SW	: comme SWY et SWE, SW est une option pour l'introduction de (). L'option est prise si plusieurs exposants ou facteurs composent respectivement un facteur ou un terme (voir <u>ANYP</u> )

Fonctions appelées

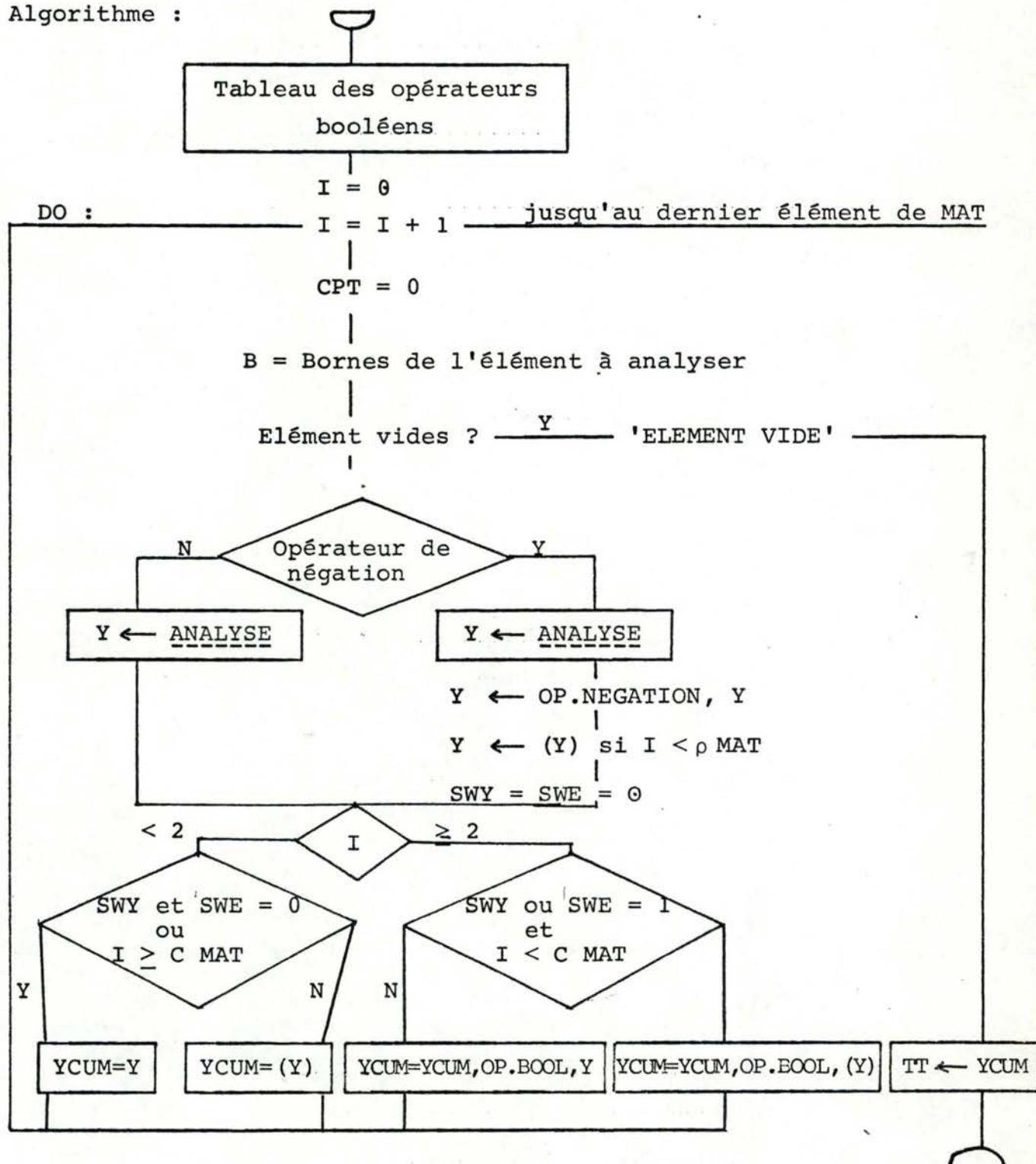
<u>TABLEAU</u>	: construit un tableau de T, F ou E pour un niveau de ()
<u>RANGE</u>	: range un tableau en fonction de l'évaluation APL par la droite
<u>TTF</u>	: (appel récursif)
<u>ANYP</u>	: enserre éventuellement un terme ou un facteur de parenthèses
<u>ANALYSE</u>	: analyse un élément de base

Fonction

TT  $\leftarrow$  LAG BOOL N

\* Objet : Analyse la structure des opérateurs booléens et de comparaison, après traduction, elle recompose l'expression de départ.

\* Algorithme :



\*

Fonctions	Variables					Param.
	INPUT		LOCALES	OUTPUT		FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>
						N LAG TT
		<input type="checkbox"/> IO				
		B XX NP OPB,OPRB Y,YCUM SWY,SWE				
	N LAG					
			BPO IDX MAT OPRO,ITF I CPT			
<u>ANALYSE</u>			(cf. <u>ANALYSE</u> )	Y		
				SWY,SWE Y YCUM		
					TT	



\* Variables utilisées

N : voir EXPR  
 LAG : retard portant sur l'expression à analyser  
 TT : résultat de la traduction  
☐ IO : index origine  
 B : Bornes de l'expression à considérer  
 XX : nom propre à APLT de l'équation à traduire  
 NP : vecteur contenant le niveau de parenthétisation de XX  
 Y, YCUM : voir EXPR  
 SWY, SWE : switch permettant de définir s'il faut ou non introduire des parenthèses  
 OPB, OPRB : vecteur des opérateurs booléens; OPRB est la représentation numérique des opérateurs booléens dans XX; BPO est la traduction APL de OPB  
 MAT : tableau des éléments composants l'expression analysée  
 I : indice courant sur MAT  
 IDX : vecteur des indices de l'expression à considérer de B[1] à B[2]  
 MAT, OPRO : (cf. TABLEAU)  
 ITF  
 CPT : indicateur de présence d'un opérateur de négation

Fonction appelée

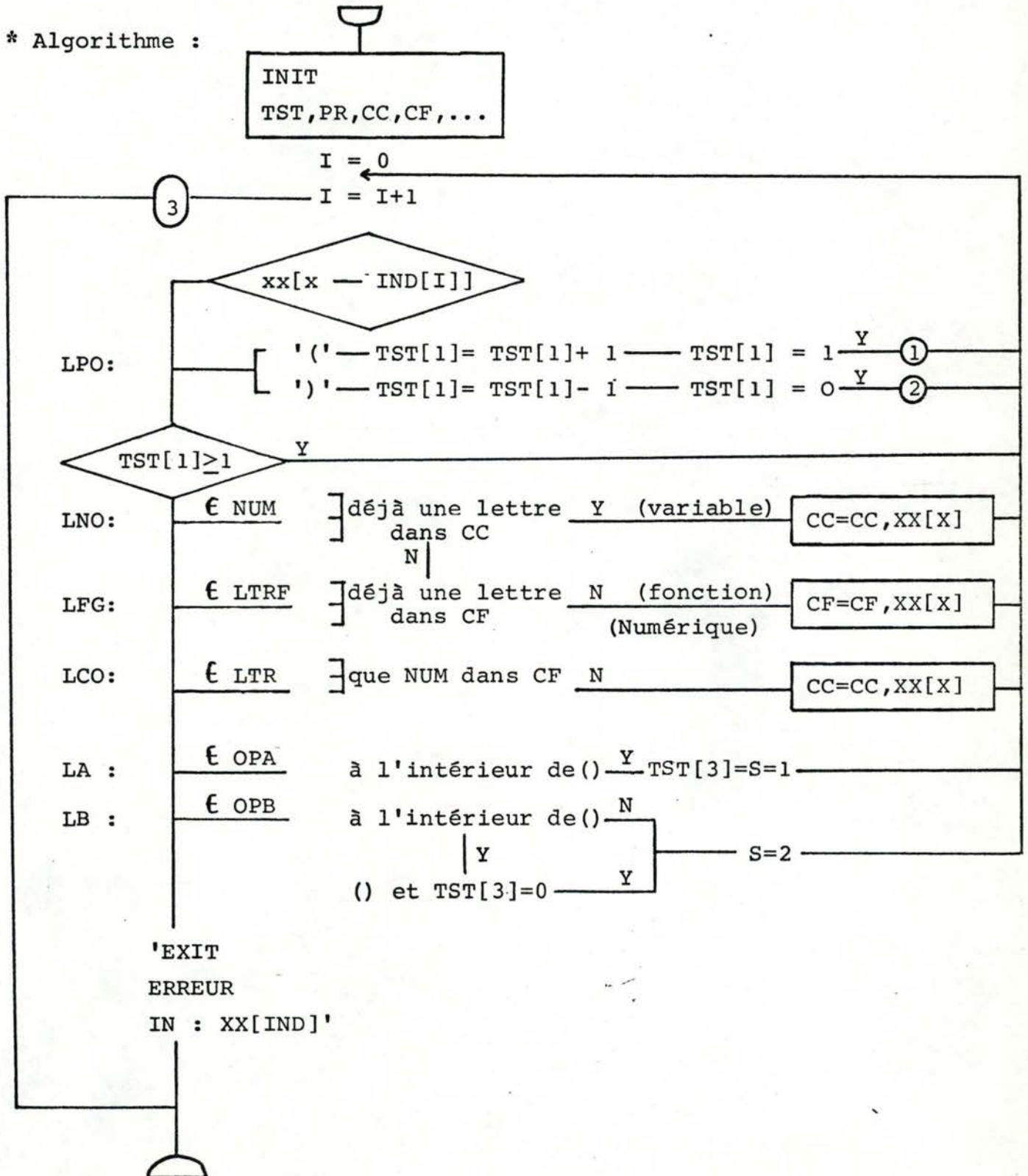
ANALYSE : analyse la composition d'un élément de base de l'expression. OP. Econométrique ? Retard ? ...

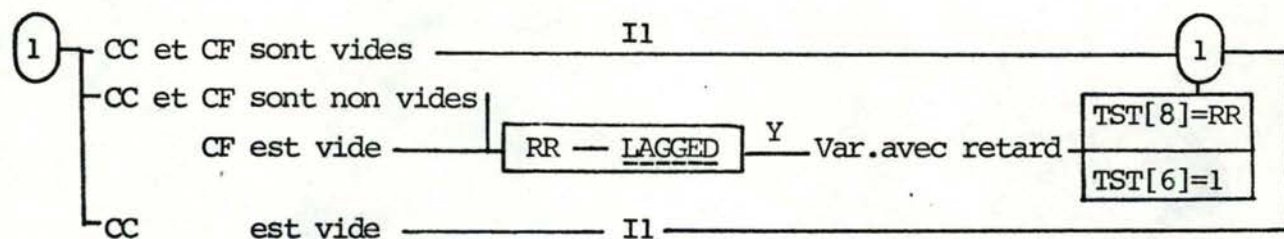
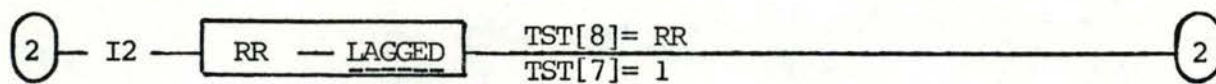
Fonction

VT  $\leftarrow$  ANALYSE

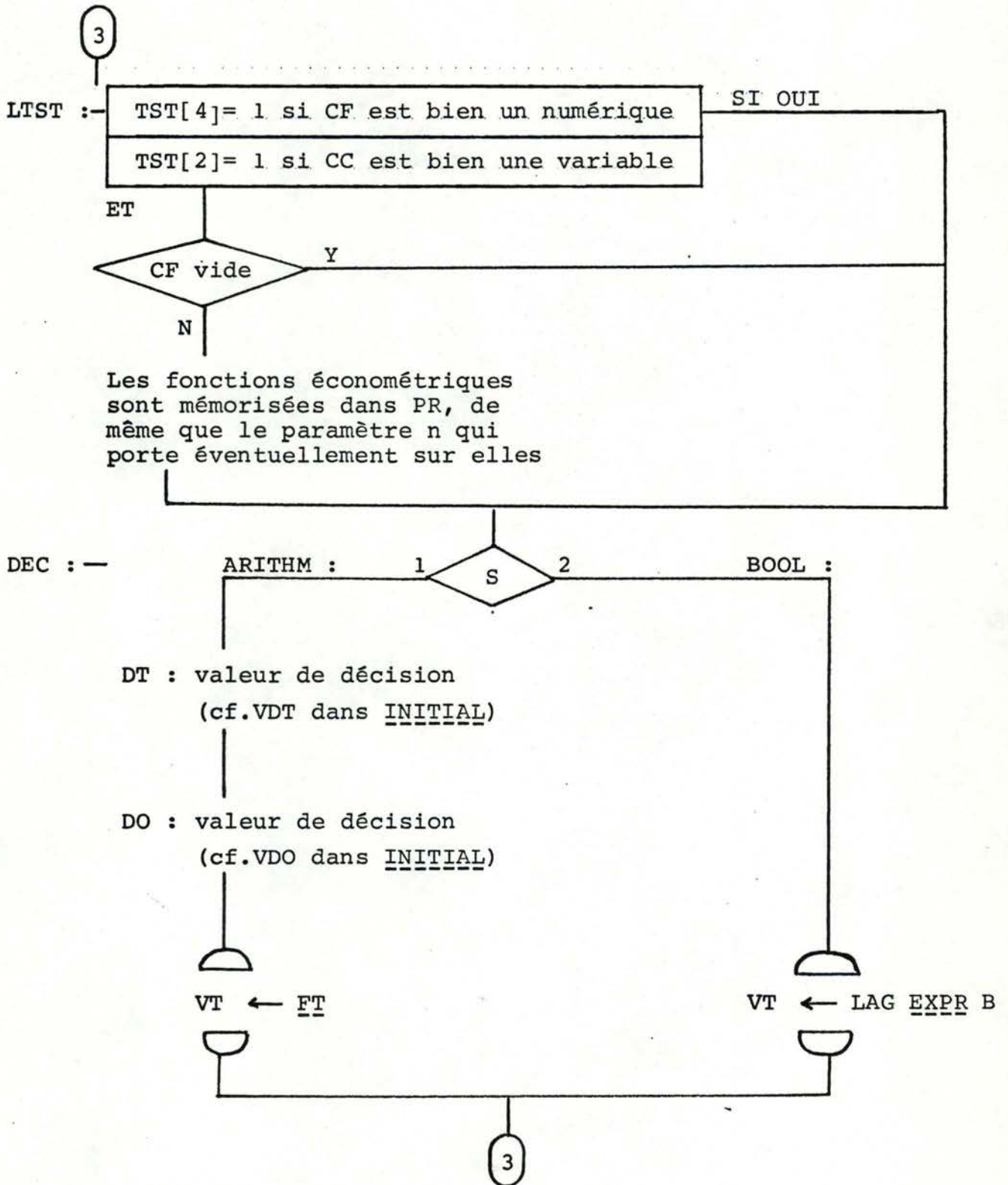
\* Objet : Analyse la composition d'un élément de base = opérateurs économétriques ? var.? exp. ?

\* Algorithme :









\*

Functions	Variables					Param.
	INPUT		LOCALES	OUTPUT		FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>
						VT
		<input type="checkbox"/> IO				
			IND TST PR			
			I, X A, K			
		S				
		NUM LTR LTRF OPA OPB	CC, CF     I1, I2			
			RR			
<u>LAGGED</u>		X NP XX NUM	cf. <u>LAGGED</u>	L	RR	
		MASK, FNT FCT STT LGT	J FC, CFFC, FCF CNUM CPT			
<u>EXPR</u>		LAG B	(cf. <u>EXPR</u> )		VT	

			DT, DO				
<u>FT</u>		DT, DO	(cf. <u>FT</u> )		VT		
					VT		



\* Variables utilisées

<input type="checkbox"/> IO	: index origine
VT	: output explicite contenant la traduction de l'élément de base (XX[IND])
IND	: portion de XX à considérer
I, X	: indices courants sur XX[IND]
TST	: vecteur de test de présence
PR	: matrice mémorisatrice de l'ordre et des paramètres n des fonctions rencontrées
A, K	: variables intermédiaires
S	: switch de présence d'une expression ARITHMETIQUE ou booléenne
CC, CF	: vecteur de mémorisation du nom d'un variable (CC) du nom des facteurs économétriques (CF) d'un numérique (CF)
NUM LTR LTRF OPA OPB	{ : (voir APLT)
I, J	: indices courants sur TCT et CF
FC, CFFC	: variables intermédiaires
FCF	: PR[2; FCF[I]] contient la valeur du paramètre n de la fonction économétrique la plus à droite, qui porte sur une variable ou sur une expression PR[2; FCF[I-1]] etc...
CNUM	: vecteur destiné à mémoriser le paramètre n d'une fonction économétrique
CPT	: variable intermédiaire
DT, DO	: valeurs de décision définissant l'élément à traduire (voir VDT, VDO dans <u>INITIAL</u> )-

Fonction appelée

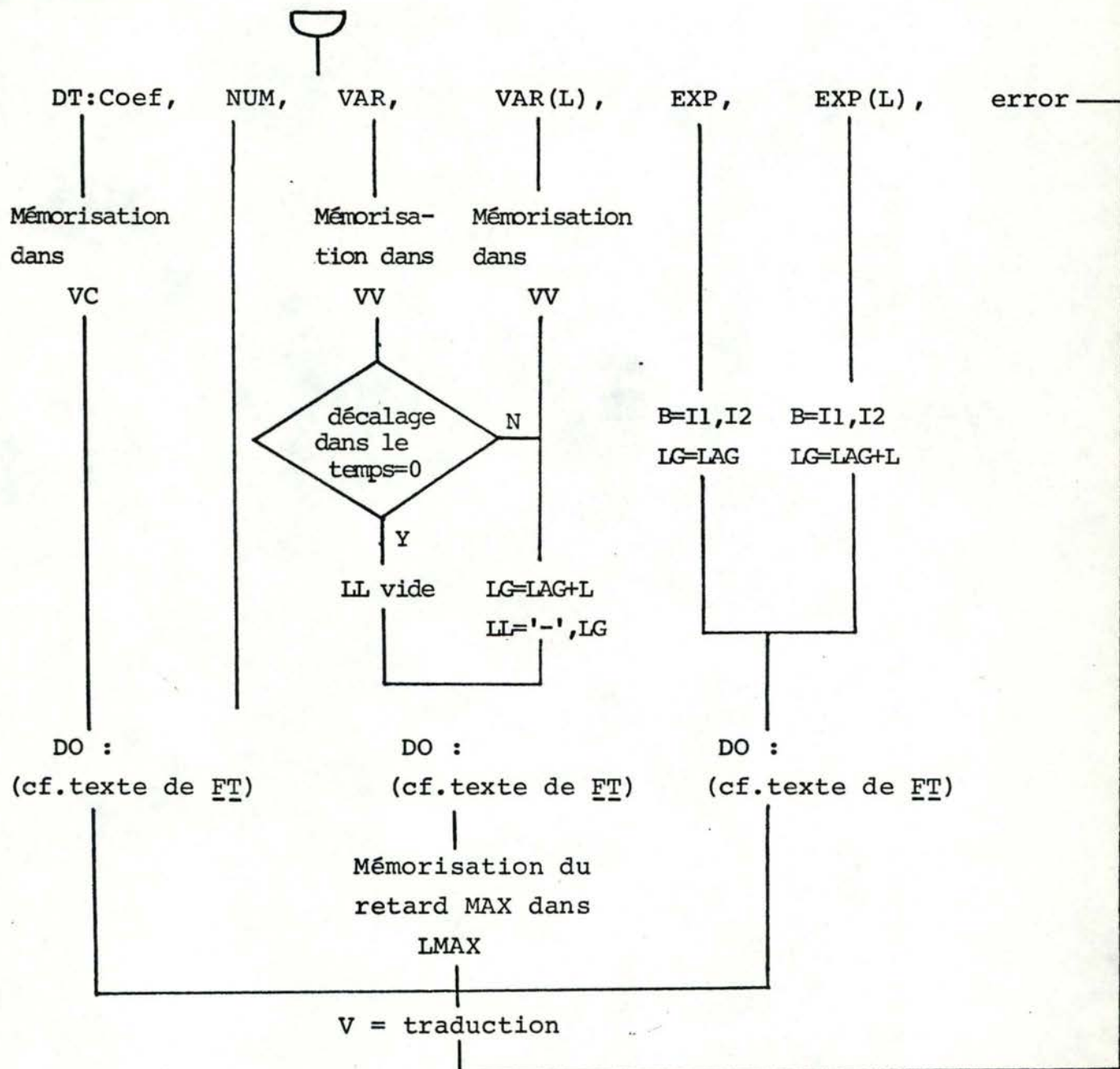
- LAGGED : fonction testant la présence d'un retard portant sur une variable ou sur une expression
- EXPR : analyse récursivement une expression arithmétique ou booléenne
- FT : génère la traduction APL en fonction de valeurs de DT et DO.

Fonction

$V \leftarrow \underline{FT}$

\* Objet : génère la traduction APL d'un élément de base en fonction de deux valeurs de décision DT et DO reçus de ANALYSE

\* Algorithme :





\*

Functions	Variables					Param.
	INPUT		LOCALES	OUTPUT M		FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>
						V
		<input type="checkbox"/> IO				
		VV,VC,IMAX				
		XX, B LAG,L,IG CF,CC,I, PR,FCF PARAM 1 2 STP 1 2 LGP 1 2 SWY,SWE VDO,VDT DO,DT				
			LL S1,S2 VO VL			
		I1,I2 NUM LTR OP	K,BB			
<u>APLPOL</u>	BB		(cf. <u>APLPOL</u> )		CC	
		CC				
<u>EXPR</u>	LG B		(cf. <u>EXPR</u> )		T	
		T				
				IMAX VU VC SWY,SWE	V	

\* Variables utilisées

V	: vecteur contenant la traduction de l'élément considéré
<input type="checkbox"/> IO	: définit l'origine des vecteurs
VV, VC, LMAX	: voir APLT
LAG, L, LG	: décalage dans le temps
L	: valeur du retard défini par l'opérateur de retard (L) (cf. <u>LAGGED</u> )
LAG	: valeur du retard s'appliquant au départ à l'expression considérée
LG	: = LAG ou = LAG + L en fonction des différents cas rencontrés
XX	: nom interne à APLT de l'équation à traduire
B	: pointeurs courants vers le début et la fin de la portion d'équation à considérer
CF, CC, I PR, FCF	: voir <u>ANALYSE</u>
PARAM 1 2	: voir <u>INITIAL</u>
STP 1 2	
LGP 1 2	
SWY, SWE	: switch permettant de définir s'il faut ou non introduire de parenthèses
	SWY = 1 : il faudrait introduire des parenthèses en raison des opérateurs unaires portant sur un variable
	SWE = 1 : il faudrait introduire des parenthèses en raison d'une expression parenthésée au départ
	"il faudrait" car cette option est à revoir en fonction de la place qu'occupe l'élément analysé dans l'expression globale (voir <u>TTF</u> )
VDO, VDT	: voir <u>INITIAL</u>
DO, DT	: valeurs de décision reçues de <u>ANALYSE</u> , et qu'il faut comparer respectivement à VDO et VDT.
LL	: variables locales permettant de simplifier l'écriture du littéral APL à générer
S1, S2	
VO	: variable locale permettant de mémoriser une étape intermédiaire de traduction

VL : variable locale de mémorisation du retard maximum appliqué à une variable

K : mémorise l'indice I de la fonction rencontrée; s'il s'agit de AV, AG, MX ou MN

I1, I2 : pointent vers le début et la fin de l'expression sur laquelle portent ces fonctions

BB : littéral de l'expression sur lequel portent ces fonctions

CC : traduction APL de BB

NUM }  
LTR } : voir APLT  
OP }

TST[5] : indique si l'argument de AV, AG, MX ou MN est effectivement une expression ou une variable simple; dans ce dernier cas, la traduction est directe sans appel à APLPOL. TST[5] marque en effet la présence ou non d'un opérateur

### Fonctions appelées

APLPOL : traduit l'expression sur laquelle porte AV, AG, MX ou MN

EXPR : analyse récursivement les différentes expressions rencontrées



Fonction

TABLEAU

\* Objet : Construit un tableau de Termes, Facteurs, Exposants pour un niveau de parenthétisation donné.

Un tableau se présente comme suit :

(1)				
				(2)

← représentation numérique du signe

← indice de début de l'élément - 1

← indice de fin de l'élément + 1

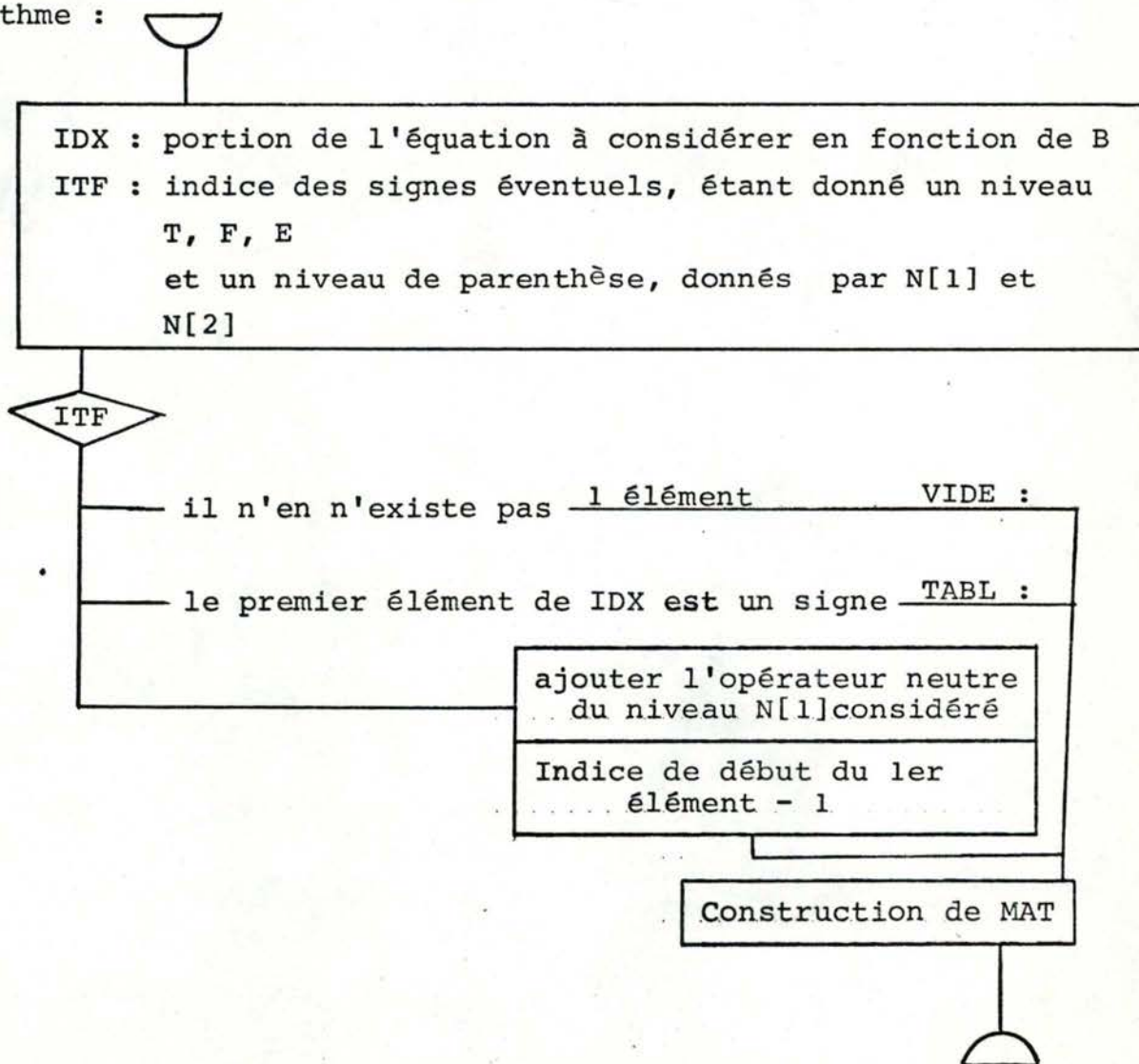
éléments: 1 2 n1 n

NB: (1) (2) bornes de l'expression à considérer


(1) = B[1] - 1

(2) = B[2] + 1

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.		
		<input type="checkbox"/> IO					
		B, N					
		NP OPR					
			IDX ITF OPRO				
				MAT			

\* Variables utilisées

<input type="checkbox"/> IO	: Définit l'origine des vecteurs ( <input type="checkbox"/> IO = 1)
B	: Bornes de la portion d'équation à considérer
N	: Vecteur représentant les niveaux auxquels on se trouve $N[1] = \begin{cases} 0 \\ 1 \\ 2 \end{cases} = \begin{matrix} \text{niveau d'un terme} \\ \text{niveau d'un facteur} \\ \text{niveau d'un exposant} \end{matrix}$ $N[2] = \text{Niveau de parenthétisation auquel on se trouve -}$
NP	: Niveau de parenthétisation pour toute l'équation
OPR	: Représentation numérique des opérateurs arithmétiques
IDX	: Vecteur valant 1 pour tous les indices compris entre B[1] et B[2]
ITF	: Vecteur contenant les indices des opérateurs arithmétique de XX (nom de l'équation à traduire) pour un niveau N[1] et N[2] donné
OPRO	: Vecteur contenant la représentation numérique des opérateurs arithmétiques trouvés et dont l'indice est dans ITF.
MAT	: tableau de Termes, Facteurs, Exposants pour un niveau N[1] et N[2] donné



Fonction

RANGE

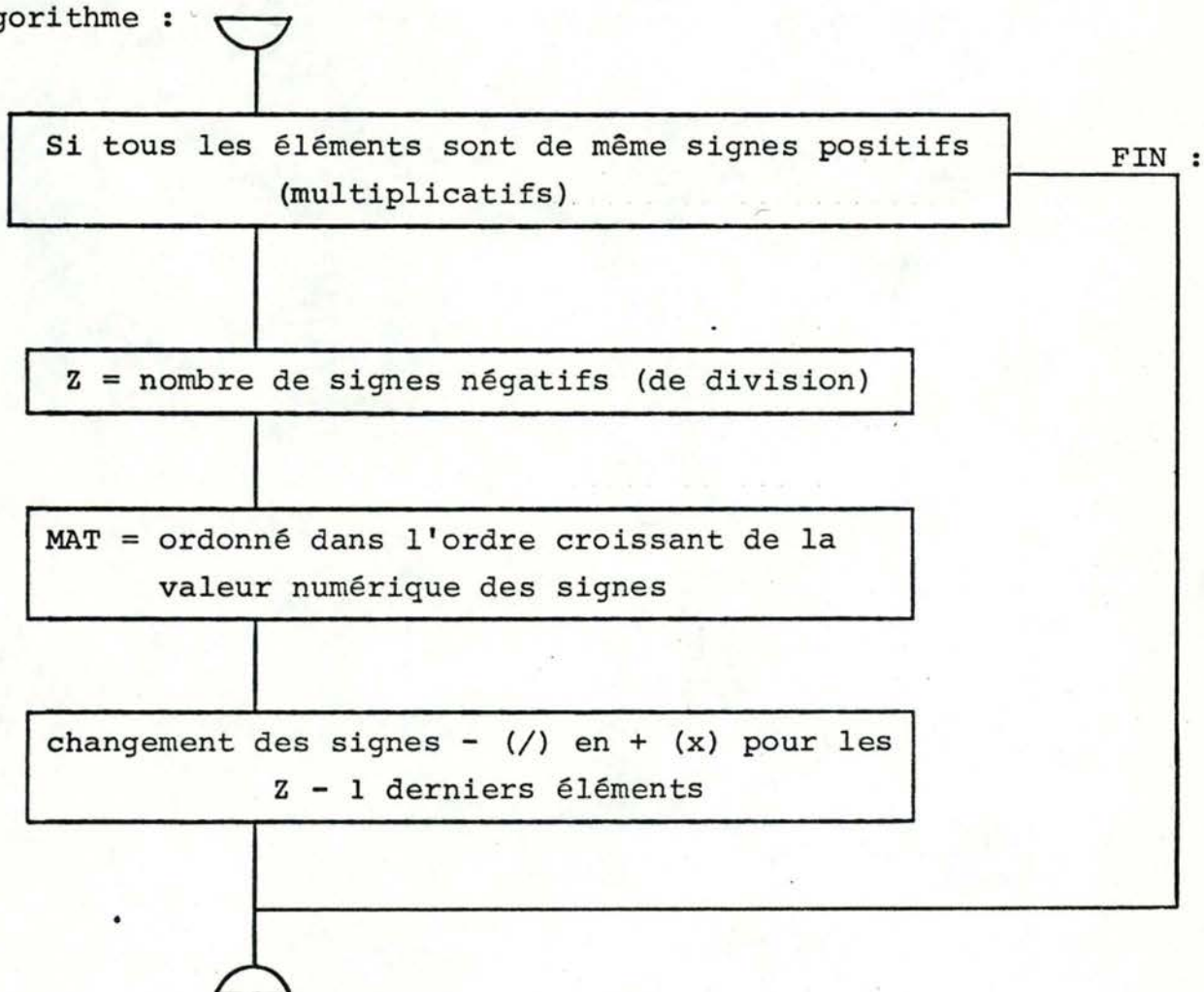
\* Objet : range les tableaux de Termes ou Facteurs en fonction de l'évaluation APL par la droite =

ex.: élément 1 - élément 2 - élément 3 + élément 4


(G - D)

= élément 1 + élément 4 - élément 2 + élément 3 (D - G)

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			
	EXPL.	IMPL.		IMPL.	EXPL.		
		<input type="checkbox"/> IO					
		N, MAT					
			A, Z				
				MAT			

\* Variables utilisées

☐ IO : définit l'origine des vecteurs ( ☐ IO = 1 )  
N : cf. EXPR  
MAT : cf. TABLEAU  
A, Z : variables de travail



Fonction

$R \leftarrow \text{LAGGED}$

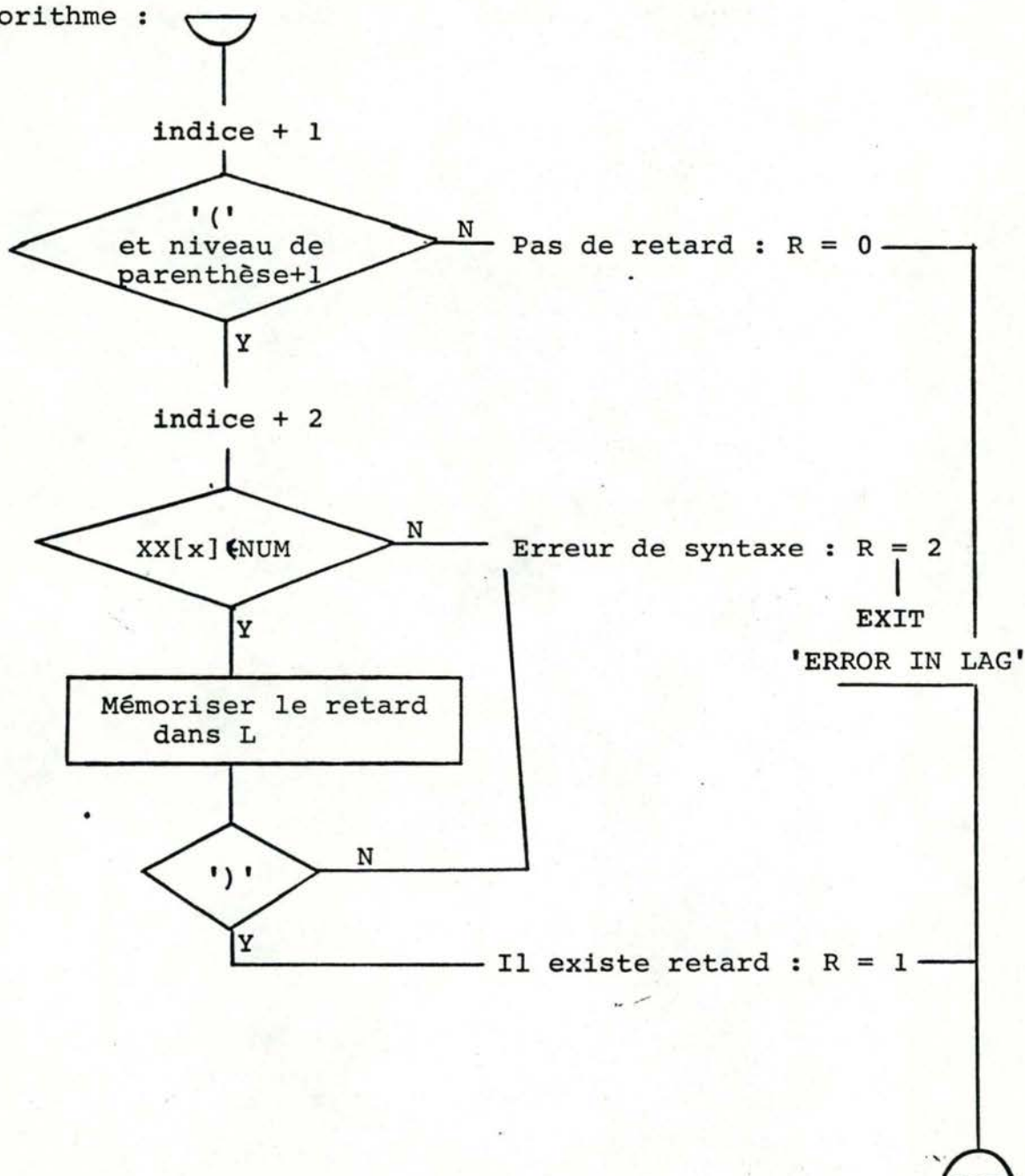
\* Objet : teste la présence d'un retard portant sur une variable ou sur une expression

R = 0 = pas de retard

= 1 = } retard; L contient la valeur du retard

= 2 = erreur de syntaxe dans le retard

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>	
		<input type="checkbox"/> IO					
		XX					
		X					
		IND					
		NUM					
		NP					
				L			
					R		

\* Variables utilisées

- ☐ IO : définit l'origine des vecteurs utilisés
- XX : nom de l'équation à traduire
- X : indice courant sur XX
- NP : vecteur contenant le niveau de parenthétisation
- NUM : ensemble des chiffres
- IND : vecteur valant 1 pour tous les indices compris entre B[1] et B[2]; il définit la portion de l'équation à considérer
- L : LAG : valeur du décalage dans le temps
- R : 0 = pas de décalage  
1 = erreur de syntaxe dans l'opérateur de retard  
2 = décalage dans le temps de valeur L

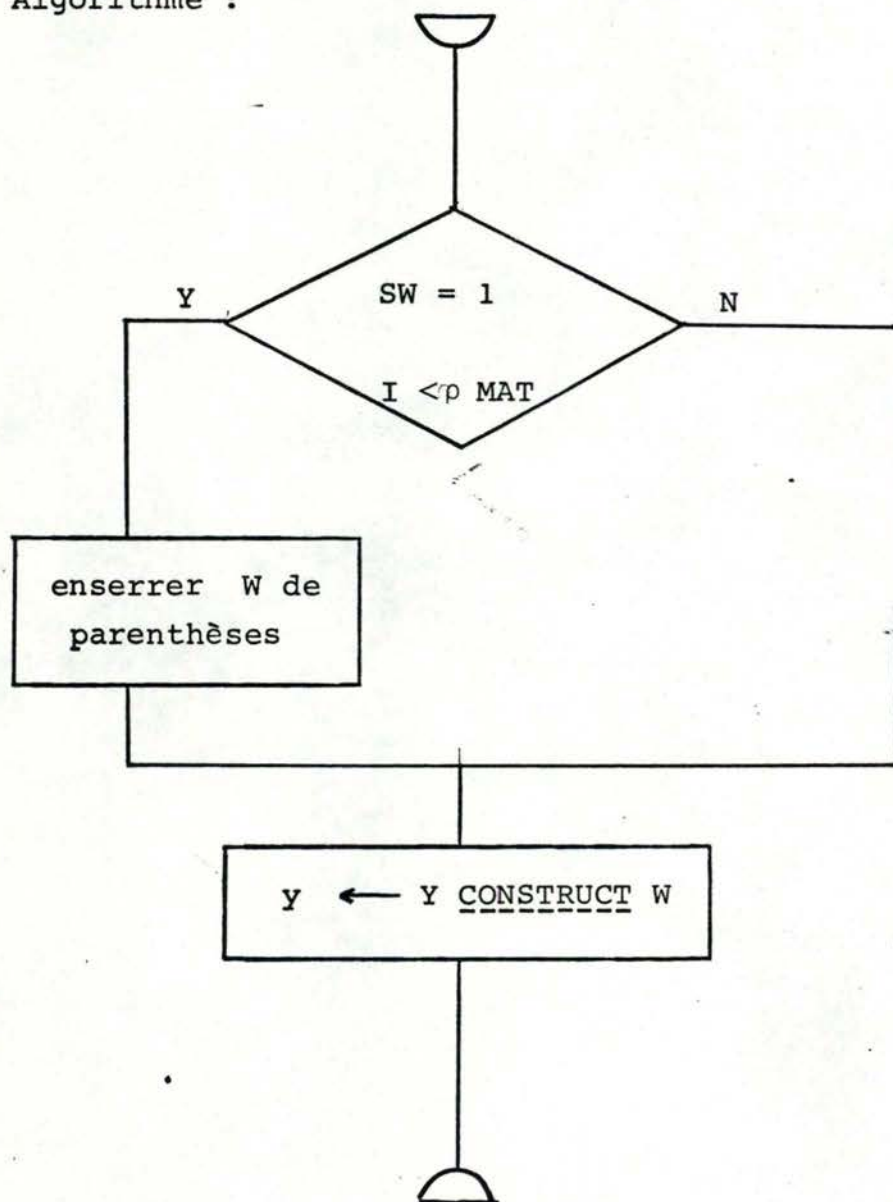


Fonction


ANYP W

\* Objet : enserrer éventuellement un terme ou un facteur de parenthèses (voir TTF)

\* Algorithme :



\*

Functions	Variables						Param.
	INPUT		LOCALES	OUTPUT			
	EXPL.	IMPL.		IMPL.	EXPL.		
	W						W
		<input type="checkbox"/> IO SW MAT I Y					
<u>CONSTRUCT</u>	Y		(cf. <u>CONSTRUCT</u> )		Y		
	W						
				W			

\* Variables utilisées

- W : nom interne de l'élément courant traduit, qu'il faut éventuellement enserrer de parenthèses
- ☐ IO : définit l'origine des vecteurs
- SW : switch indiquant s'il faut enserrer de parenthèses; ce switch est défini dans TTF
- MAT : tableau des termes, facteurs ou exposants
- I : indice courant sur MAT
- Y : vecteur contenant l'élément courant traduit

Fonctions appelées

CONSTRUCT : construit la traduction en fonction des opérateurs binaires

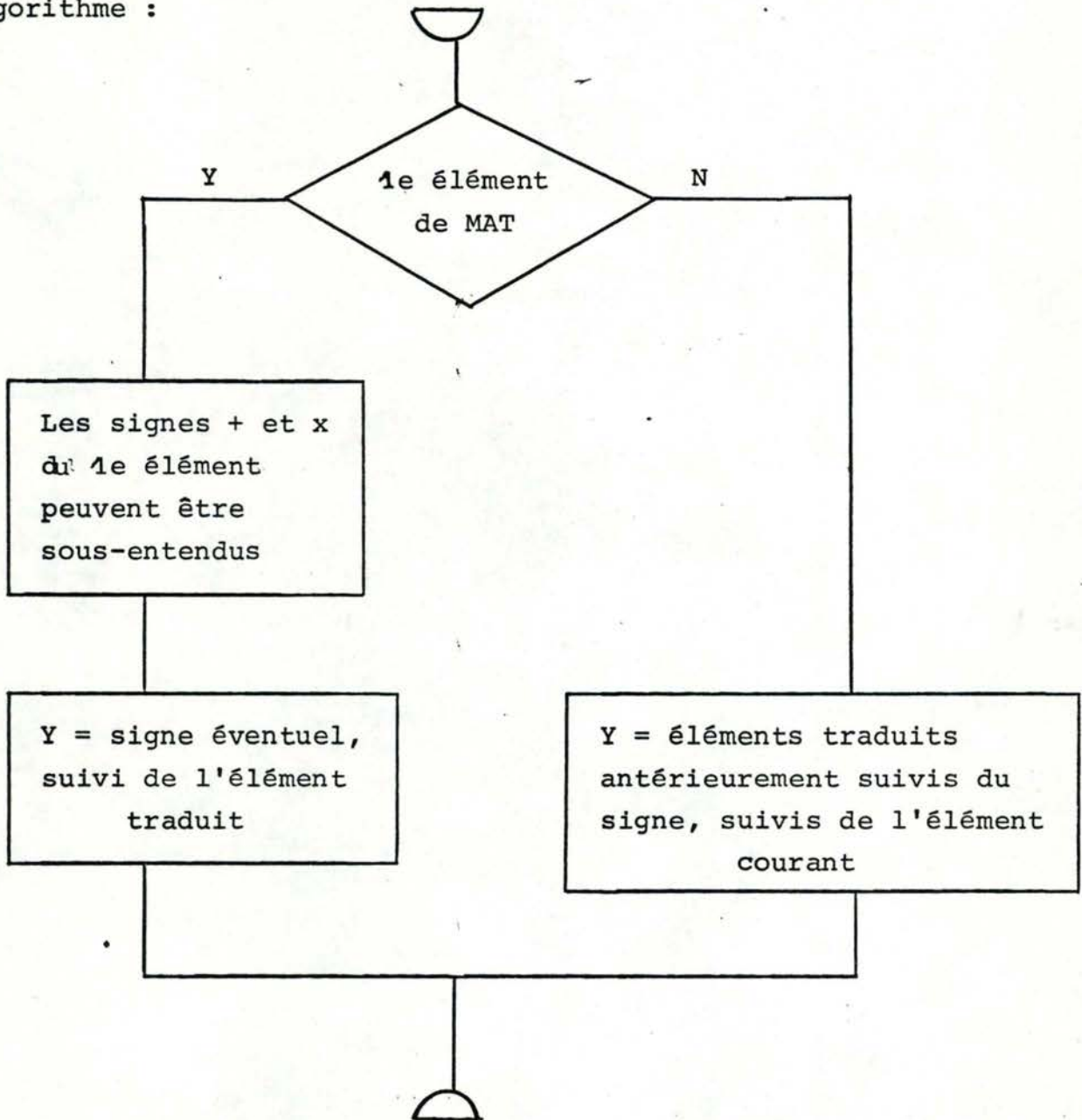


Fonction

$$Y \leftarrow YANC \text{ CONSTRUCT } X$$

\* Objet : construit la traduction en fonction des opérateurs binaires mémorisés lors de l'analyse dans MAT

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>	
	X YANC						X YANC
							Y
		I MAT OP					
					Y		

\* Variables utilisées

X : nom interne de l'élément traduit courant

YANC : nom interne des éléments traduits antérieurs

MAT : tableau des termes, facteurs, exposant

I : indice courant sur MAT

OP : liste des opérateurs binaires APL

Y : vecteur contenant le résultat de la combinaison des traductions.



Fonction

RES ← APLPOL BB

- \* Objet : les fonctions AV AG MX MN peuvent porter sur une variable ou sur une expression. Dans ce dernier cas, on se trouve devant un problème de traduction différent de celui envisagé jusqu'ici; il s'agit en effet de générer seulement une nouvelle série sur laquelle portera la fonction.

La traduction en syntaxe APL se présentera toujours sous forme d'un appel à une fonction écrite par ailleurs, mais avec comme argument une expression dont il faut noter l'absence de l'indice de temps.

On demande par exemple le taux de croissance moyen entre les 10 dernières périodes du P.N.B. à prix constants étant donné une série  $X_1$  = PNB à prix courant

et une série  $X_2$  = Indice des prix

LEC

APL

10 AG ( $X_1/X_2$ )

≡

(AG ← 10 AG  $X_1 \div X_2$ )

donnera

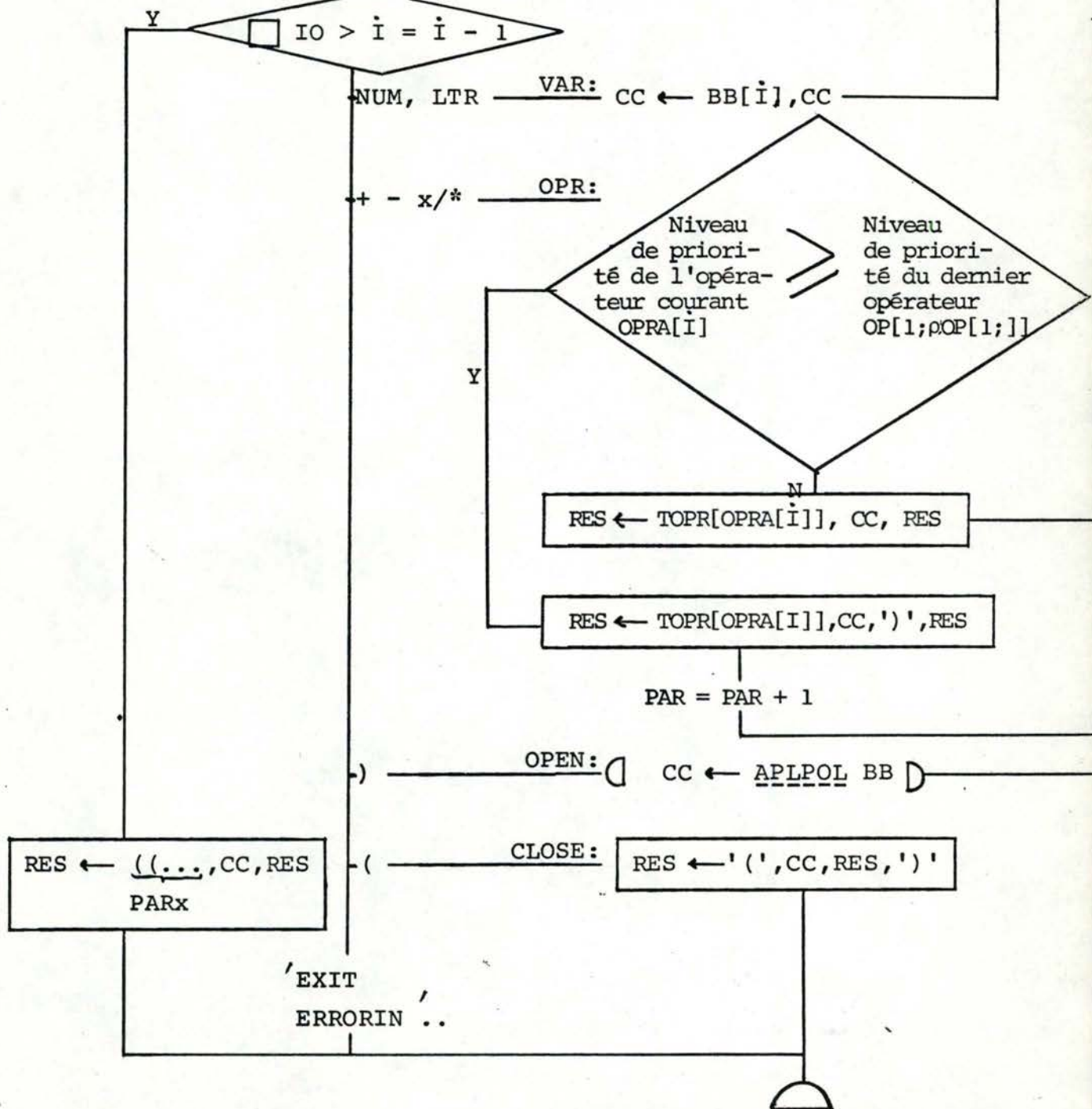
le taux de croissance moyen demandé.

\* Algorithme :

NB :

*	5
/	4
X	3
-	2
+	1

OP, PAR  
 CC, RES  
 OPRA : représentation numérique des opérateurs dans BB  
 TOPR : traduction APL des opérateurs



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>	
							BB RES
		<input type="checkbox"/> IO					
	BB	I					
			OP OPRA TOPR CC PAR				
		NUM LTR					
<u>APLPOL</u>	BB		(cf. <u>APLPOL</u> )		CC		
		CC					
				I	RES		



\* Variables utilisées

BB : littéral de l'expression à traduire  
 I : indice courant sur BB  
 RES : traduction APL de BB  
☐ IO : définit l'origine des vecteurs ( ☐ IO = 1)  
 OP : mémorise les opérateurs rencontrés  
 OPRA : représentation numérique des opérateurs de BB  
 TOPR : traduction APL des opérateurs  
 CC : variable locale de mémorisation  
 PAR : nombre de parenthèses qu'on ouvre (quand on rencontre  
       ')') et qu'il faudra fermer à la fin  
 NUM { :  
 LTR { : voir APLT

Fonctions appelées

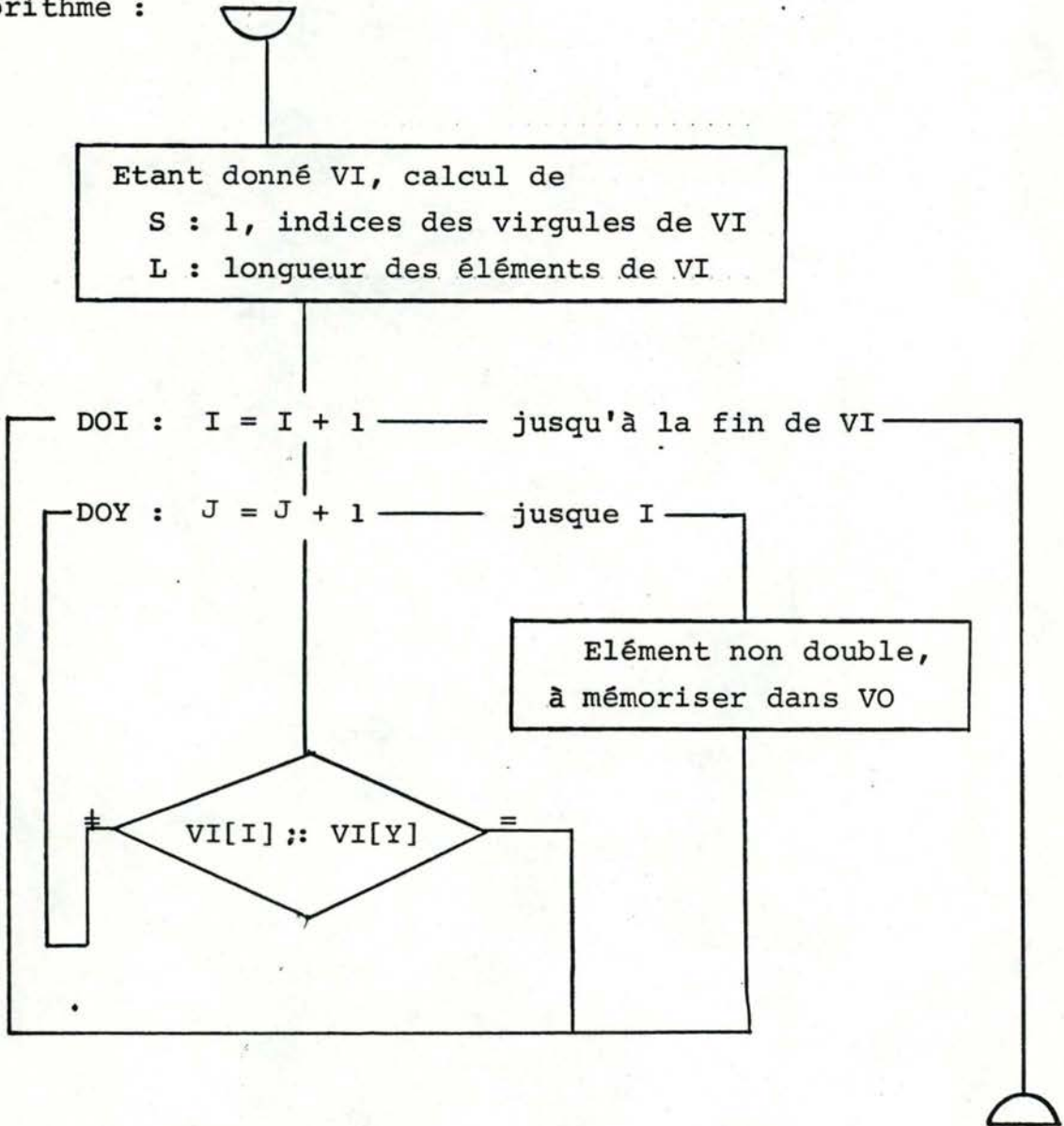
APLPOL : appel récursif.

Fonction

VO  $\leftarrow$  ARIEL VI

\* Objet : nettoie VI de tous ses éléments doubles, chaque élément VI étant séparé par une virgule

\* Algorithme :



\*

Fonctions	Variables						Param.
	INPUT		LOCALES	OUTPUT			FORMELS
	EXPL.	IMPL.		IMPL.	EXPL.	<input type="checkbox"/>	
	VI						VI
							VO
		<input type="checkbox"/> IO					
			S L I Y				
					VO		



\* Variables utilisées

VI : vecteur Input duquel il faut supprimer les éléments en double

VO : vecteur Output = VI dont on a supprimé les éléments doubles

☐ IO : définit l'origine des vecteurs ( ☐ IO = 1 )

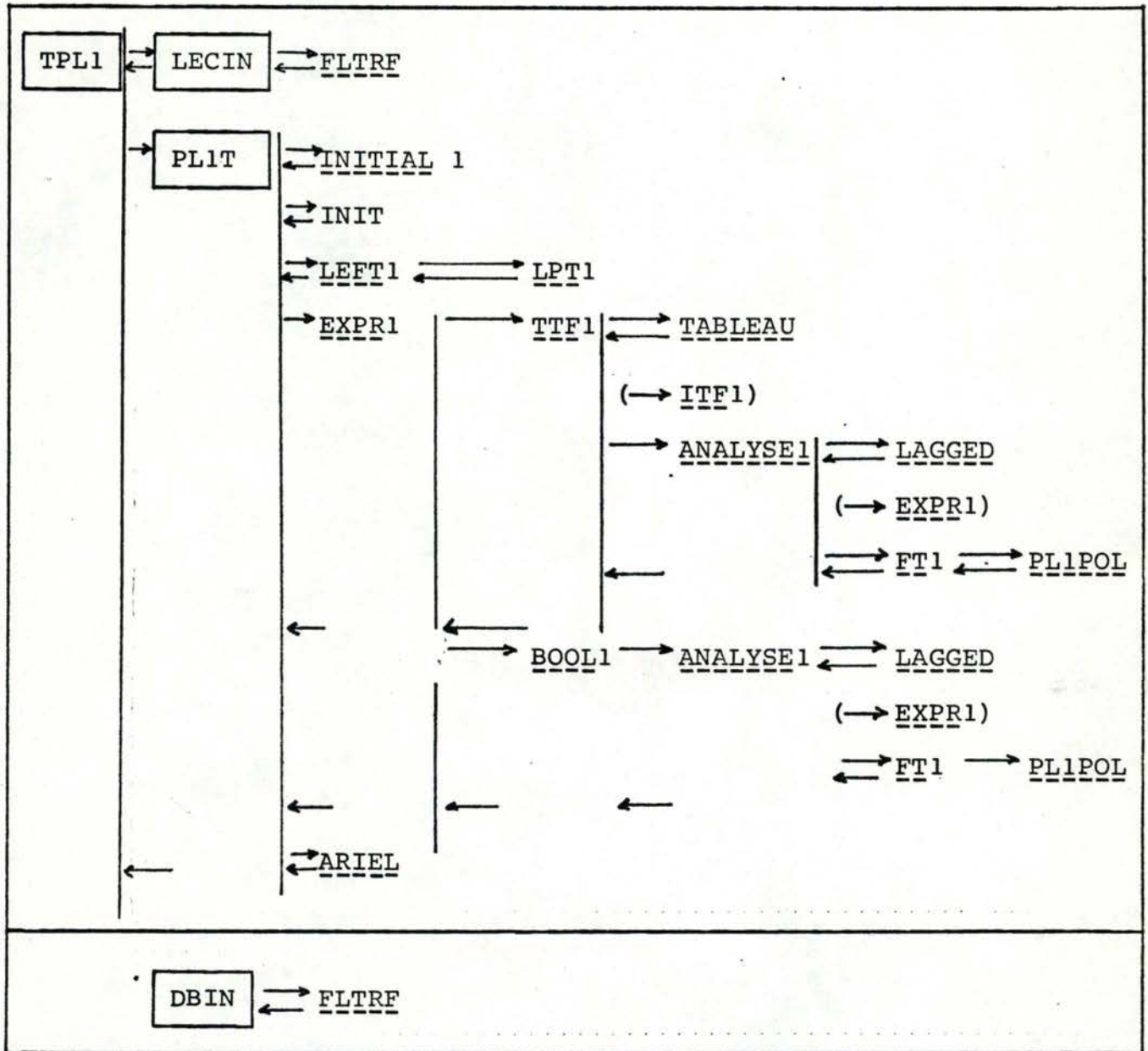
S : vecteur d'indices des différents éléments de VI;  
les virgules de VI

L : vecteur des longueurs des différents éléments de VI

I, J : indices courant sur VI

## 2. TPL1GP

L'enchaînement des différentes fonctions du groupe TPL1GP se schématise comme suit :



où - → représente l'appel d'une fonction

- ← représente le retour à la fonction appelante

- (→) représente un appel récursif

- □ représente une fonction directement utilisable par l'utilisateur

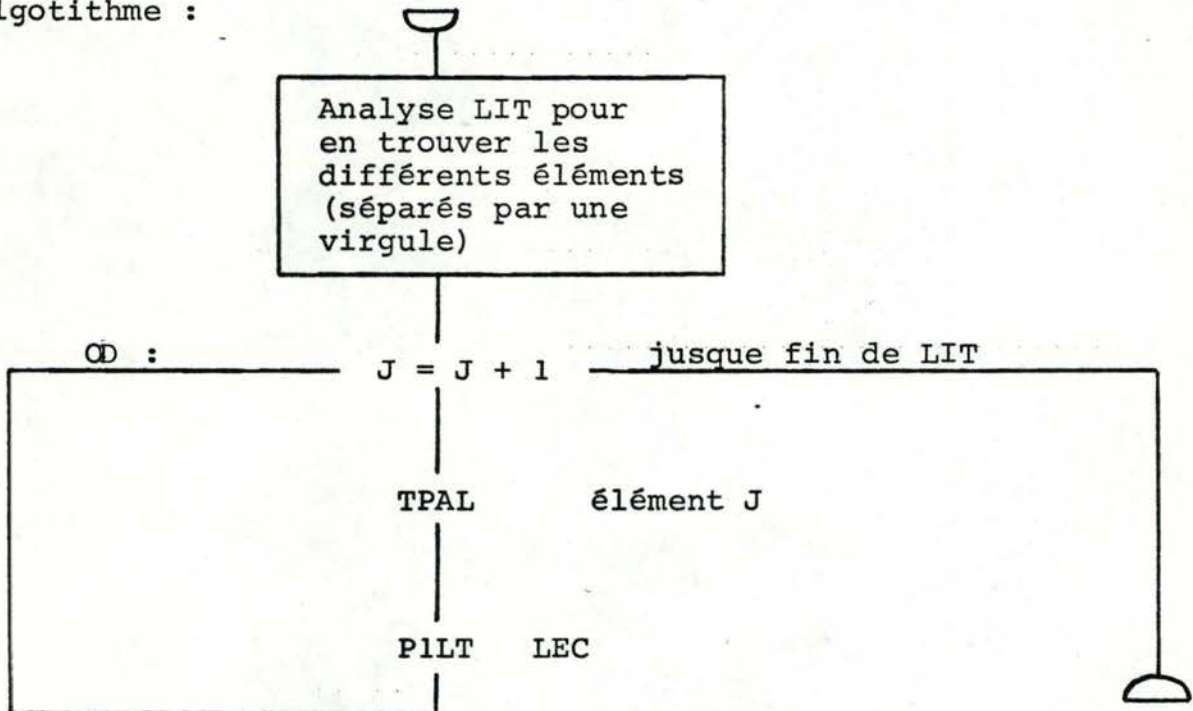
Les algorithmes des différentes fonctions de ce groupe sont similaires à ceux de TAPLGP.

## 3. TRADGP

La fonction principale de ce groupe est TRAD LIT

\* Objet : elle traduit les noms d'équations contenus dans LIT, en APL et PL1

\* Algorithme :



Les deux fonctions TPAL YY et PL1T XX sont équivalents aux

fonctions TAPL YY et PL1T XX sauf que pour TPAL :

- elle imprime en plus le nom de l'équation à traduire
- elle crée les variables
 

'A'NOM-EQ	(cf. DOSSIER Utilisation de TRADGP)
'V'NOM-EQ	
'C'NOM-EQ	
'L'NOM-EQ	

pour PL1T :

- elle crée les variables
 

'P'NOM-EQ	(cf. DOSSIER Utilisation de TRADGP)
'C1'NOM-EQ	

Les autres fonctions de ce groupe sont celles de TAPLGP et TPL1GP.



00000000

```

▽ APLT XX;IO:LTF:ITFF:NUM:CAR:NP:LPC:EQ:PETARD:BOFNES:OPR:OP:OPA:OPR:OPRR:F
CT:STT:IGT:PARAM1:STP1:IGP1:PARAM2:STP2:IGP2:FNT:MASK:SW:SWY:SWE:Y:V:T
[1] LTF←'ABCDEFGHIJKIMNOPQRSTUVWXYZ'
[2] ITFF←'ABCDEFGHIJKIMNOPQRSTUVWXYZ'
[3] NUM←'0123456789.'
[4] CAR←'+-x÷*( )=<=>≠VΛ~'
[5] [←0p[←LEC←XX,0o[←'LEC:',0o[←'*****',0o[←'LEC FORM:',0o[←APL←0pS←IO←1
[6] INITIAL
[7] OPA←1+OPA←'+-x÷*'
[8] OP←'+-x÷*'
[9] OPR←'<=>≠VΛ~'
[10] RSTOP1←IGOP1←0pOP1←OPR
[11] →EXIT×11=INIT
[12] BOFNES←0,0XX
[13] →EXIT×11=LEFT
[14] VC←' ',Y←0pSW←LMAX←PETARD←0
[15] OPR←'+-x÷*'1XX
[16] OPRR←OPR1XX
[17] →EXIT×11=Λ/' '=EQ←PETARD EXPP BOFNES
[18] VV←1+VV←APIEL VV
[19] VC←1+VC←APIEL VC
[20] →0,0o[←IMAX←VIMAX,0o[←'IMAX:',0o[←VC,0o[←'VC:',0o[←VV,0o[←'VV:',0o[←APL←ZZ
    ,EQ,0o[←'APL:',0o[←'*****',0o[←'EQ. IN APL FORM
[21] EXIT:[←'ERROR IN LEC FORM '

```

▽

00000000

```

▽ DP←DPIN LEC:I:CF:T:TST:STTST:IGTST:TPAD:STTPAD:IGTPAD:A;R;PET:STPPT:IGPPT;
J
[1] LEC←(1-LECε' ')/LEC,0oIO←1
[2] TST←'CDPLADPDLEDPFLDLPAVAGMYMN'
[3] STTST← 0 1 2 3 4 5 7 9 11 13 15 17 19 21 23
[4] IGTST← 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
[5] TRAD←'C:DIF:RAP:LOG:ABS:DEL:FAP:DEL:LOG:RAP:DEL:RAP:LOG:LOG:DEL:LOG:RAP:AV:
    AG:MAX:MIN:'
[6] STTRAD← 0 2 6 10 14 18 26 34 42 50 58 66 69 72 76
[7] IGTRAD← 2 4 4 4 4 8 8 8 8 8 8 3 3 4 4
[8] A←'+-x÷*( )=<=>≠VΛ~'
[9] PRT←' + - * / ** .LT. .LE. .EQ. .GE. .GT. .NE. .AND. .OR. .NO
    ,
[10] STPPT← 0 3 6 9 12 16 22 28 34 40 46 52 59 65 71
[11] IGPPT← 3 3 3 3 4 6 6 6 6 6 6 7 6 6 1
[12] CF←DP←0pI←T←0
[13] DO:→0×1(0LEC)<I←I+1
[14] →INL×1LEC[I]ε'1234567890.ABCDEFGHIJKIMNOPQRSTUVWXYZ()'
[15] →LIF×1LEC[I]ε'ABCDEFGHIJKIMNOPQRSTUVWXYZ'
[16] →LOC,0oP←P/10P←AεLEC[I]
[17] EXIT:→0,0o[←'EXIT DPIN',LEC[1I]
[18] INL:→DO,0oCF←0oDP←DP,(V←ELTFE CF),LEC[I],0oJ←0
[19] LIF:→DO,0oCF←CF,LEC[I]
[20] LOC:→LOC8×1(P=8)ΛT=0

```

21] →DO,0oCF+0oDB+DP,PPT[STPRT[E]+1LGPRT[B]]  
 22] LOC8:→DO,0oCF+0oDB+DE,' = ',0oT+1

▽

00000000

▽ LEC←LECIN DR;TST;STTST;IGTST;TRAD;STTRAD;LGTRAD;NUM;LTR;S;A;M;V;J;AST;I  
 1] NUM←'1234567890'  
 2] ITP←'ABCDEFGHIJKIMNOPQRSTUVWXYZ'  
 3] TST←'CDIFFAPLOGARSAVAGMAXMINLTLEOEGCTNEANDORNO'  
 4] STTST← 0 1 4 7 10 13 15 17 20 23 25 27 29 31 33 35 38 40  
 5] IGTST← 1 3 3 3 3 2 2 3 3 2 2 2 2 2 2 3 2 2  
 6] TRAD←'CDEIAAVAGMYMN<=>>#AV~'  
 7] STTRAD← 0 1 2 3 4 5 7 9 11 13 14 15 16 17 18 19 20 21  
 8] LGTRAD← 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1  
 9] LEC←M+0oI+0oDP+DP,' '  
 10] DO1:→DO1×1[VO>I+I-1,0oLEC+DP[I],LEC  
 11] DO2:→DO1×1DE[I]∈LTP,MUM,'()=+-'  
 12] →LTST×1V/A+\*.:/'∈DE[I],0oAST+0  
 13] EXIT:→0,LEC+0o[+ ' IN ',DP,0o[+ 'EPPFUR : ',DP[1I],0o[+ '  
 14] LTST:→(LAST,LPT,I2PT,LDIV)[A+A/10A]  
 15] LDIV:→EXIT×10≥I+I-1  
 16] →DO2,0oLEC+\*',LEC  
 17] LAST:→LAST×1S,0oAST+AST+S+DP[I+I-1]∈\*'  
 18] →(LA1,LA2,EXIT)[(1×AST=0)+(2×AST=1)+3×AST>1]  
 19] LA1:→DO2,0oLEC+\*',LEC  
 20] LA2:→DO2,0oLEC+\*',LEC  
 21] LPT:→EXIT×10≥I+I-1  
 22] →LPT×1DP[I]∈' '  
 23] →EXIT×1~V/A+(DE[I]∈NUM),(DE[I]∈LTR),(DE[I]∈\*+~/x'),DE[I]∈(''  
 24] →(LPN,LPL,LPS,LPP)[A/10A]  
 25] LPP:→DO1,0oLEC+\*',LEC  
 26] LPN:→EXIT×10≥I+I-1,0oLEC+DP[I],LEC  
 27] →LPN×1DP[I]∈NUM  
 28] →DO2  
 29] LPL:→EXIT×10≥I+I-1,0oM+DP[I],M  
 30] →LPL×1DP[I]∈ITP  
 31] →EXIT×1(~DP[I]∈'.')v0=0M  
 32] →(0=0V+ELTPE M)0EXIT,0oJ+9  
 33] →DO1,0oI+I-1,M+0oLEC+V,LEC  
 34] LPS:→DO1,0oLEC+\*',LEC,0oI+I+1  
 35] I2PT:→EXIT×10≥I+I-1  
 36] →I2PT×1DP[I]∈' '  
 37] PT2:→I×1DP[I]∈ITP  
 38] →EXIT×10=0M  
 39] →(0=0V+ELTPE M)0EXIT,0oJ+0  
 40] →D1,M+0oLEC+V,LEC  
 41] I:→(VO>I+I-1)0FIN,0oM+DP[I],M  
 42] →PT2  
 43] D1:→I2PT×1DP[I]∈':'  
 44] →DO1  
 45] FIN:→EXIT×10=0M  
 46] →(0=0V+ELTPE M)0EXIT,0oJ+0  
 47] M←0oLEC+V,LEC  
 48] OD1:LEC←1+LEC



V

00000000

```

V PL1T XY;PIO;LTP;LTPF;NUM;CAP;NP;IPC;EQ;RETARD;FORNES;OPR;OP;OPA;OPR;OPRR;F
CT;STT;IGT;PAPAM1;STP1;LGP1;PARAM2;STP2;LGP2;FNT;MASK;SW;SWY;SWE;Y;T;V
1] LTP←'ABCDEFGHIJKL MNOPQRSTU VWXYZ'
2] LTR←'ABCDEFGHIJKL MNOPQRSTU VWXYZ'
3] NUM←'0123456789.'
4] CAP←'+-x:*( )=<<>>≠VA~'
5] [←0p[←LEC+XX,0o[←'LEC:',0p[←'*****',0o[←'LEC FORM:',0o[←',0pPAR←1-S←[
IO←1
6] INITIAL1
7] OPA←1+OPA←'+-x:~'
8] OP←'+-*/'
9] OPB←'<=>≠VA~'
10] OP1←'<=>≠~|'
11] STOP1← 0 1 3 4 6 7 9 10 11
12] LGOP1← 1 2 1 2 1 2 1 1 1
13] →EXIT×11=INIT
14] FORNES←0,0XX
15] →EXIT×11=LEFT1
16] VC←' ',Y←0oSW←IMAX←RETARD←0
17] OPR←'+-x:~'1XX
18] OPRR←OPR1XX
19] →EXIT×11=^(EQ+(EQ+RETARD FYPE1 FORNES),PARo'))ε'Y'
20] VV←1+VV←ARIEL VV
21] VC←1+VC←ARIEL VC
22] →0,0o[←IMAX←IMAX,0o[←'IMAX:',0p[←VC,0o[←'VC:',0o[←VV,0o[←'VV:',0o[←PL1←ZZ
,EO,0o[←'PL1:',0o[←'*****',0o[←'EQ. IN PL1 FORM
23] EXIT:[←'ERROR IN LEC FORM'

```

V

00000000

```

V PL1T XY;PIO;LTP;LTPF;NUM;CAP;NP;IPC;EQ;RETARD;FORNES;OPR;OP;OPA;OPR;OPRR;F
CT;STT;IGT;PAPAM1;STP1;LGP1;PARAM2;STP2;LGP2;FNT;MASK;SW;SWY;SWE;Y;T;V
1] LTR←'ABCDEFGHIJKL MNOPQRSTU VWXYZ'
2] LTR←'ABCDEFGHIJKL MNOPQRSTU VWXYZ'
3] NUM←'0123456789.'
4] CAP←'+-x:*( )=<<>>≠VA~'
5] PAR←1-S←PIO←1
6] INITIAL1
7] OPA←1+OPA←'+-x:~'
8] OP←'+-*/'
9] OPB←'<=>≠VA~'
10] OP1←'<=>≠~|'
11] STOP1← 0 1 3 4 6 7 9 10 11
12] LGOP1← 1 2 1 2 1 2 1 1 1
13] →EXIT×11=INIT
14] FORNES←0,0XX
15] →EXIT×11=LEFT1
16] VC←' ',Y←0oSW←IMAX←RETARD←0
17] OPR←'+-x:~'1XX
18] OPRR←OPR1XX

```



```

9] →EXIT×11=Λ/(EQ←(EQ←RETARD EXPR1 FORNES),PARρ'))ε') '
0] VV←1+VV+ARIEL VV
1] VC←1+VC+ARIEL VC
2] □←PL1+ZZ,EQ,0o□←'PL1:',0o□←'*****',0o□←'EQ. IN PL1 F
    ORM FOR SIMUL : '
3] →0,0o□←'P',Q,←'PL1',0o□←'C1',Q,←'VC'
4] EXIT:□←'ERPOR IN LEC FORM '

```

▽

\*\*\*\*\*

```

▽ TAPL YY;□IO
] □←YY,0o□←'DRE:',0o□←'*****',0o□←'EQ.TO TRADUCT : ',0o□←0o□IO+1
] →(0=0LEC←LECIN YY)00
] APLT LEC
] DRE←YY

```

▽

\*\*\*\*\*

```

▽ TPAL YY;□IO;APL;VV;VC;LMAX
] □←YY,0o□←'DRE:',0o□←'*****',0o□←'EQ.TO TRADUCT : ',Q,0o□←0o□IO+1
] →(0=0LEC←LECIN YY)00
] APLT LEC
] →(0=0APL)0FIN,0oDRE←YY
] →0,0o□←('A',Q),←'APL',0o□←('V',Q),←'VV',0o□←('C',Q),←'VC',0o□←('L',Q),←'LMAX'
] FIN:→

```

▽

\*\*\*\*\*

```

▽ TPL1 YY;□IO
] □←YY,0o□←'DRE:',0o□←'*****',0o□←'EQ.TO TRADUCT : ',0o□←0o□IO+1
] →(0=0LEC←LECIN YY)00
] PL1T LEC
] DRE←YY

```

▽

\*\*\*\*\*

```

▽ TPAN LIT;Z;I;J;S;L;Q
] →T×11=0I←(S←□IO,(LITε',')/1oLIT←' ',LIT,','),0oJ←I+0
] DO:→((0S)≤I+I+1)00D
] →DO,0oL[I]+S[I+1]-S[I]+1
] OD:→((0L)≤J+J+1)00
] TED:Q←Z+LIT[S[J]+1L[J]]
] TPAL(Z+2Z)
] PL1T LEC
] →OD
] T:→TED,0oJ←1,0oL←0LIT

```

▽

\*\*\*\*\*

```

▽ AG←N AG X;Y

```

```

0  →ST×1(N<0Y)∧N>0,00±'Y+',VX
1  N←(0Y)-1
2  ST:AG←(Y[0Y]÷Y[(0Y)-N])×1÷N

```

▽

000000

```

▽ VT←ANALYSE;IND;TST;DT;DO;PR;I1;I2;CC;CF;CNUM;PR;I;J;CPT;FC;CFEC;FCF;A;X;K
1  IND←R[1]+1P[2]-R[1]
2  TST←800
3  A TST[18]:(),VAR,OPA,NUM,ε(,VAR(,EXP,LCORRECT
4  PR←(2,0STT)00
5  CC←CF+CNUM+00FNT+MASK+I+J+CPT+1-S+1
6  LOOP:→LTST×1(0IND)<I+I+1
7  →LP0×1V/A+()'εXX[X+IND[I]]
8  →LOOP×1TST[1]≥1
9  →LNO×1XX[X]εNUM
10 →LF0×1XX[X]εLTRF
11 →LC0×1XX[X]εLTR
12 →LA×1XX[X]εOPA
13 →LP×1XX[X]εOPB,00K+XX[IND]
14 EXIT:→0,VT←00[←' IN ',XX[IND],00[←'ERREUR : ',K
15 LNO:→LF0×10=0CC
16 →IN1×1CC[1]εLTR
17 LF0:→EXIT×1(S=1)∧V/CCεLTR,00K+CF,' ',CC
18 →LOOP,00CF+CF,XX[X]
19 IN1:→LOOP,00CC+CC,XX[X]
20 LC0:→EXIT×1(0≠0CF)∧∧/CFεNUM,00K+CF
21 →LOOP,00CC+CC,XX[X]
22 LP0:→(I1,I2)[A+A/10A]
23 IA:→EXIT×1TST[1]=0,00S+TST[3]+1,00K+XX[IND]
24 →LOOP
25 IP:→LOOP×1~(TST[1]=0)∨(TST[1]=1)∧TST[3]=0
26 →LOOP,CC+00S+2
27 I1:→LOOP×11<TST[1]+TST[1]+1
28 →II×1~∧/A+(0=0CC),0=0CF
29 I11:→LOOP,00I1+Y-1-TST[5]+1
30 LL:→EXIT×1V/(CC,CF)ε'.',00K+XX[IND]
31 →I00×1∧/A=0
32 →(I10,I01)[A/10A]
33 L00:→EXIT×1((0≠0CF)∧∧/CFεNUM)∨(1+CF)εNUM,00K+CF
34 L01:→EXIT×11≠PR+LAGGED,00X+X-1,00K+XX[IND]
35 →LOOP,00TST[1]+TST[1]-1,00I+I+(0V/L)+TST[6]+1
36 L10:→EXIT×1((0≠0CF)∧∧/CFεNUM)∨(1+CF)εNUM,00K+CF
37 →L11
38 L2:→LOOP×10<TST[1]+TST[1]-1
39 →LOOP×1IND[0IND]=X,00I2+X-TST[7]+1
40 →EXIT×12=FR+LAGGED,00K+XX[IND]
41 →LOOP,00TST[8]+FR,00I+I+3×FR
42 LTST:→DEC×1TST[4]+(0≠0CF)∧∧/CFεNUM,00TST[2]+(0≠0CC)∧(∧/CCεLTR,NUM)∧(TST[6]=
    0)∧TST[7]=0
43 →DEC×10=0CF
44 DO1:→DEC×11>FNT+FNT-1
45 DO2:→DO1×1∧/' '=FC,00CFEC+CFεFC+ECT[STT[J]+1LGT[J+J+1]],00I+(0CF)-FNT
46 DO3:→DO2×1(FCF+I+1FNT)[1]<1

```



```

] →DN1×1Λ/CFEC[FCF]
] →DO3,00I+I-1
] DN1:→MEM×11>ECF[1]+ECF[1]-1
] →MEM×1~CF[ECF[1]]∈NUM
] →DN1,00CNUM+CF[ECF[1]],CNUM
] MEM:→EXIT×1V/CC∈',',00K+CC
] →EXIT×1V/(CNUM+0',CNUM)∈',',00K+CF
] PR[;J]+(CPT+ECF[1]+1),0CNUM,00K+CF
] →DO2×10≠0CF,CNUM+00CF+(ECF[1]+CF),(ECF[1]+FNT+(0CNUM)-1)+CF
] DEC:→(DEC1,DEC2)[S]
] DEC2:→0,00VT+LAG EXP B
] DEC1:DT+TST[2]+(2×TST[4])+(4×TST[6])+(8×TST[7])+16×TST[8]
] I+0ECF+ECF×PP[1;ECF+APR[1;]]≠0,00PP[2;DECT]+PR[2;DECT]+PP[2;DECT]=0
] →EXIT×1V/PP[2;NECT]≠0
] DO+ECF[I]+(ECF[I-1]×10×2)+(ECF[I-2]×10×4)+ECF[I-3]×10×6
] VT+ET

```

V

210100

```

V VT+ANALYSE1:IND:TST:DT;DO;PR;I1;I2;CC;CF;CNUM;PP;I;J;CPT;EC;CFEC;ECF;A;X;K
IND+P[1]+1P[2]-P[1]
TST+800
P TST[18]:(),VAR,OPA,NUM,∈(,VAR(,EXP,LCORRECT
PR+(2,0STT)00
CC+CF+CNUM+00FNT+MASK+I+J+CPT+1-S+1
LOOP:→ITST×1(0IND)<I+I+1
→LPO×1V/A+()'∈XX[X+IND[I]]
→LOOP×1TST[1]>1
→LNO×1XX[X]∈NUM
→LFO×1XX[X]∈LTPF
→LCO×1XX[X]∈LTR
→LA×1XX[X]∈OPA
→LBR×1XX[X]∈OPR,00K+XX[IND]
EXIT:→0,VT+00□+ ' IN ',XX[IND],00□+'ERREUR : ',K
LNO:→LFO×10=0CC
→LN1×1CC[1]∈LTR
LFO:→EXIT×1(S=1)ΛV/CC∈LTP,00K+CF,' ',CC
→LOOP,00CF+CF,XX[X]
LN1:→ICOP,00CC+CC,XX[X]
LCO:→EXIT×1(0≠0CF)ΛΛ/CF∈NUM,00K+CF
→LOOP,00CC+CC,XX[X]
LPO:→(I1,I2)[A+A/10A]
LA:→EXIT×1TST[1]=0,00S+TST[3]+1,00K+XX[IND]
→LOOP
LP:→LOOP×1~(TST[1]=0)V(TST[1]=1)ΛTST[3]=0
→LOOP,CC+00S+2
I1:→LOOP×11<TST[1]+TST[1]+1
→LI×1~Λ/A+(0=0CC),0=0CF
I11:→ICOP,00I1+X-1-TST[5]+1
LI:→EXIT×1V/(CC,CF)∈',',00K+XX[IND]
→LOO×1Λ/A=0
→(LI0,LI1)[A/10A]
LOO:→EXIT×1((0≠0CF)ΛΛ/CF∈NUM)V(1+CF)∈NUM,00K+CF
LO1:→EXIT×11≠PP+LAGGED,00X+X-1,00K+XX[IND]

```



```

] →LOOP,0oTST[1]+TST[1]-1,0oI+I+(pVL)+TST[6]+1
] L10:→EXIT×1((0≠oCF)∧∧/CF∈NUM)∨(1+CF)∈NUM,0oK+CF
] →L11
] L2:→LOOP×10<TST[1]+TST[1]-1
] →LOOP×1IND[oIND]=X,0oI2+Y-TST[7]+1
] →EXIT×12=FF+LAGGED,0oK+XX[IND]
] →LOOP,0oTST[8]+FF,0oI+I+3×FF
] L1TST:→DEC×1TST[4]+(0≠oCF)∧∧/CF∈NUM,0oTST[2]+(0≠oCC)∧(∧/CC∈LTR,NUM)∧(TST[6]=
0)∧TST[7]=0
] →DEC×10=0CF
] DO1:→DEC×11>FNT+FNT-1
] DO2:→DO1×1∧/' '=FC,0oCFEC+CF∈FC+ECT[STT[J]+1IGT[J+J+1]],0oI+(oCF)-FNT
] DO3:→DO2×1(FCF+I+1FNT)[1]<1
] →DN1×1∧/CFEC[FCF]
] →DO3,0oI+I-1
] DN1:→MEM×11>FCF[1]+FCF[1]-1
] →MEM×1~CF[FCF[1]]∈NUM
] →DN1,0oCNUM+CF[FCF[1]],CNUM
] MEM:→EXIT×1∨/CC∈' ',0oK+CC
] →FYIT×1∨/(CNUM+0',CNUM)∈' ',0oK+CF
] PR[;J]+(CPT+FCF[1]+1),CNUM,0oK+CF
] →DO2×10≠oCF,CNUM+0oCF+(FCF[1]+CF),(FCF[1]+FNT+(oCNUM)-1)+CF
] DEC:→(DEC1,DEC2)[S]
] DEC2:→0,0oVT+LAG FYPE1 B
] DEC1:DT+TST[2]+(2×TST[4])+(4×TST[6])+(8×TST[7])+16×TST[8]
] I+oFCF+FCF×PR[1;FCF+PR[1;]]≠0,0oPR[2;DECT]+PR[2;DECT]+PR[2;DECT]=0
] →EXIT×1∨/PP[2;NFACT]≠0
] DO+FCF[I]+(FCF[I-1]×10×2)+(FCF[I-2]×10×4)+FCF[I-3]×10×6
] VT+ET1
]

```

▽

000000

```

▽ ANYP W
] →(0=SW×(I<oMAT[1;]))oCST
] W←('W,')
] CST:Y+Y CONSTRUCT W
]

```

▽

000000

```

▽ RES+APIEOL FP:OPRA;CC;OP;PAR;TOPP
] CC←RES+0oOP+(2 1)o(8+PAR+0),I
] OPRA←'+-×÷*'1PB
] TOPP←'+-×÷*'
] DO:→OP×1[IO>I+I-1
] →VAR×1BP[I]∈NUM,LTR
] →OPR×1BP[I]∈'+-×÷*'
] →OPEN×1BP[I]∈')'
] →CLOSE×1BP[I]∈'('
] EXIT:→0,0o[←'ERROR IN : ',BB
] VAR:→DO,0oCC+BP[I],CC
] OPR:→ST×1([OPRA[I]÷2]>[OP[1;OP[1;]]÷2
] →DO,CC+0oRES+TOPP[OPRA[I]],CC,RES,0oOP+OP,[2] OPRA[I],I
] ST:→DO,CC+0oRES+TOPP[OPRA[I]],CC,')',RES,0oOP+OP,[2] OPRA[I],I,0oPAR+PAR+1
]

```

```

OD:→0,0pRES+(PAP0('),CC,RES
OPEN:→DO,0pCC+APIEOL PP
CLOSE:RES+('),CC,RES,')'

```

▽

```

=====

```

```

▽ VO+APIEL VI;S;L;I;J
L←S+[IO,(VIε',')]/1pVI
→0×11=0VI,0pVO+',' ,0oI+0
DO:→((0S)≤I+I+1)0OD
→DO,0pL[I]+S[I+1]-S[I]+1
OD:VO+',' ,VI[S[I]+1L[I+1]]
DOI:→((0L)≤I+I+1)0oJ+0
DOJ:→(I≤J+J+1)0ODJ
→(L[I]≠L[J])0DOJ
→(0=Λ/VI[S[I]+1L[I]]=VI[S[J]+1L[J]])0DOJ
→DOI
ODJ:→DOI,0oVO+VO,',' ,VI[S[I]+1L[I]]

```

▽

```

=====

```

```

▽ AV←N AV Y;Y
→ST×1(N<0Y)ΛN>0,0o2'Y+',' ,Y
N←oY
ST:AV←(+/(0Y)-N)+Y)÷N

```

▽

```

=====

```

```

▽ TT+LAG POOL N;I;MAT;IDX:ITF:OPRO;RPO;CPT
RPO←'>=<=≤>~',0oI+0
→EXIT×10=0ITF+((NP[IDX]=N[2])ΛOPRE[IDX]<0OPB)/IDX+B[1]+1B[2]-B[1]
→EXIT×1ITF[1]=IDX[1]
OPRO←0,OPRE[ITF]
ITF←(IDX[1]-1),ITF
MAT←(3,0ITF)0OPRO,ITF,1+ITF,B[2]+1
MAT←Φ[2] MAT
MAT[1;]←1ΦMAT[1;]
DO:→((0MAT[2;])<I+I+1)0OD,0pCPT+0
P←MAT[2;I],MAT[3;I]-1
→LT×1P[1]≠P[2]
→OD,0o[←'ELEMENT VIDE : ',XX[1B[1]+1]
LT:→D1×1~OPRE[B[1]+1]=0OPB
→IT,0oCPT+CPT+1,0oP[1]+P[1]+1
D1:Y←(((2|CPT)ΛI<0MAT[1;])0'('),((2|CPT)0'~'),Y+ANALYSE,((2|CPT)ΛI<0MAT[1;]
)0'')'
→D2×1~I<2
→D21×1(CPT=0)∨(0=SWYvSWE)∨I<0MAT[1;]
Y←(' ,Y,')'
D21:→DO,0oYCUM←Y
D2:→D3×1(SWYvSWE)Λ(I<0MAT[1;])Λ(CPT=0)ΛI<0MAT[1;]
→DO,0oY+YCUM←YCUM,RPO[MAT[1;I]],Y
D3:→DO,0oY+YCUM←YCUM,((MAT[1;I]≠0)0RPO[MAT[1;I]]),(' ,Y,')'

```



OD:→0,0pTT+Y  
EXIT:□+ 'BOOLEAN EXIT : ',XX[IDX]

▽

□□□□

▽ TT+LAG POOL1 N:I;MAT:IDX:ITF:OPPO

I←0  
→EXIT×10=0ITF+((NP[IDX]=N[2])^OPPP[IDX]<0OPP)/IDX+P[1]+1P[2]-P[1]  
→EXIT×1ITF[1]=IDX[1]  
OPPO←0,OPPP[ITF]  
ITF←(IDX[1]-1),ITF  
MAT←(3,0ITF)0OPPO,ITF,1+ITF,P[2]+1  
DO:→((0MAT[2:])<I+I+1)0OD,0oCPT+0  
P←MAT[2:I],MAT[3:I]-1  
→IT×1P[1]≠P[2]  
→OD,0o□+ 'ELEMENT VIDE : ',XX[1P[1]+1]  
IT:→D1×1~OPPP[P[1]+1]=0OPP  
→IT,0oCPT+CPT+1,0oP[1]+P[1]+1  
D1:Y←((2|CPT)0OP1[STOP1[0STOP1]+1IGOP1[0STOP1]]),Y+ANALYSE1  
→D10×1~SWYε1,2,3  
Y←(' ',Y,')'  
D10:→D2×1~I<2  
→DO,0oYCUM←Y  
D2:→DO,0oY←YCUM+YCUM,OP1[STOP1[MAT[1:I]]+1IGOP1[MAT[1:I]]],Y  
OD:→0,0pTT+Y,0oSWE←1  
EXIT:□+ 'BOOLEAN EXIT : ',XX[IDX]

▽

□□□□

▽ Y+YANG CONSTRUCT X

→(I=1)0FIRSTI  
→(1,0oY+YANG,OP[MAT[1:I]],X)00  
FIRSTI:→(MAT[1:I]ε(2 4))0LSIGN  
→(1,0oY+X)00  
LSIGN:Y←OP[MAT[1:I]],X,0oSWE←MAT[1:1]=4

▽

□□□□

▽ EXP+LAG EXPP P:ECUM:ECUM:F:F:YCUM:IG:FONCTION

ECUM←ECUM+F+F+YCUM+FONCTION+0oIG+I+0  
→POOLN×1S=2

ARITH:→0,0oEXP+LAG TTE(0,NP[R[1]])

POOLN:EXP+LAG POOL(0,NP[R[1]])

▽

□□□□

▽ EXP+LAG EXPR1 P:ECUM:ECUM:F:F:YCUM:IG:FONCTION

ECUM←ECUM+F+F+YCUM+FONCTION+0oIG+L+0

→POOLN×1S=2

ARITH:→0,0oEXP+LAG TTE1(0,NP[R[1]])

POOLN:EXP+LAG POOL1(0,NP[R[1]])



7

7 VT←PLTFF CC  
 →ST×10=0CC  
 DO:→ST×1(0STTST)←J+J+1  
 →DO×1(0TST[STTST[J]+1LGTST[J]])≠0CC  
 →DO×1~^/TST[STTST[J]+1LGTST[J]]=CC  
 →0,0oVT←TRAD[STTPAD[J]+1LGTRAD[J]]  
 ST:VT←'

7

7 V←ET;VL;LL;S1;S2;V0;K;BB;TOPR  
 DET:→DET+VDT1DT  
 CO:→EXIT+VDO1DO  
 N2:→SWY+0,CF+0oV+CF  
 V1:→(EXIT+VDO1DO)×1LAG=IG+0,0oLL+0oVV+VV,CC,',',0oSWY+1  
 V4:→EXIT+VDO1DO,0oLL←'-',VIG+LAG+L,0oVV+VV,CC,',',0oSWY+1  
 F8:→EYP+VDO1DO,0oR←I1,I2,0oIG+LAG,0oSWY+0  
 F24:→EYP+VDO1DO,0oR←I1,I2,0oIG+LAG+L,0oSWY+0  
 EXIT:→0,0o[←'COMBINAISON DE FONCTIONS NON ADMISE PORTANT SUR : ',CC,V+'  
 V:→IM,0oV+CC,['I',LL,']',0oSWY+0,0oVL+IG  
 C:→CORF  
 D:→IM,0oV+CC,['I',LL,']-','CC,['I-',(VVL),']',0oVL+PR[2;FCF[I]]+IG  
 E:→IM,0oV+CC,['I',LL,']÷','CC,['I-',(VVL),']',0oVL+PR[2;FCF[I]]+IG  
 L:→IM,0oV+@','CC,['I',LL,']',0oVL+IG  
 A:→IM,0oV+|','CC,['I',LL,']',0oVL+IG  
 DR:→IM,0oV+(',CC,['I',LL,']÷','CC,['I-',(VPR[2;FCF[I]]+IG),'])-','CC,['I-',(VPR[2;FCF[I-1]]+IG),']÷','CC,['I-',(VVL+PR[2;FCF[I]]+PR[2;FCF[I-1]]+LG)  
 PD:→IM,0oV+(',CC,['I',LL,']-','CC,['I-',(VPR[2;FCF[I]]+IG),'])-','CC,['I-',(VPR[2;FCF[I-1]]+LG),']-','CC,['I-',(VVL+PR[2;FCF[I]]+PR[2;FCF[I-1]]+LG)  
 LD:→IM,0oV+@','CC,['I',LL,']-','CC,['I-',(VVL+PR[2;FCF[I]]+IG),']  
 LE:→IM,0oV+@','CC,['I',LL,']÷','CC,['I-',(VVL+PR[2;FCF[I]]+IG),']  
 DL:→IM,0oV+(@','CC,['I',LL,']-@','CC,['I-',(VVL+PR[2;FCF[I-1]]+LG),']  
 RL:→IM,0oV+(@','CC,['I',LL,']÷@','CC,['I-',(VVL+PR[2;FCF[I-1]]+LG),']  
 AVR:→APLPOL×1TST[5],0oK+I  
 OK:→0,0oV+(',PARAM1[STP1[DO]+1LGP1[DO]],(VPR[2;FCF[K]]),PARAM2[STP2[DO]+1LGP2[DO]],CC,')'  
 AG:→AVR  
 MX:→AVR  
 MN:→AVR  
 EXP:→0,0o[←'COMBINAISON DE FONCTIONS NON ADMISE PORTANT SUR : ',XX[(I1-2)+1I2-I1-1],V+'  
 E:→LS1,0oS1←'  
 CF:→EXP  
 DE:→LL11,0oS1←'-',S2←'  
 EE:→LL11,0oS1←'÷',S2←'  
 LE:→LS1,0oS1←@'  
 AE:→LS1,0oS1←|'  
 DEE:→LS12,0oS1←'-',0oS2←'÷'  
 EDE:→LS12,0oS1←'÷',0oS2←'-'  
 LDE:→LL22,0oS1←@',0oS2←'-'

```

LFE:→LL22,0pS1←'⊙',0pS2←'÷'
DLE:→LL11,0pS1←'-' ,0pS2←'⊙',0pI←I-1
ELE:→LL11,0pS1←'÷',0pS2←'⊙',0pI←I-1
AVE:→AVR
AVE:→AVF
AGE:→AVR
MYE:→AVF
MNE:→AVR
→EXP
COEF:→EXIT×1~(Λ/CFεNIM)Λ0≠0CF
→SWY←0,CF←0pVC+VC,(V←'C[' ,CF,' ]'),', '
APLPOL:→OK,0pCC←APLPOL RR,0pI←1+0pPR+XX[I1+I2-I1]
IS1:→(1=Λ/' '=T+IG EXPP R)00
→0,0pV←S1,T,0pSWE←1
IS12:→(1=Λ/' '=T+IG EXPP R)00
V0←(' ,T,' )',S2
→(1=Λ/' '=T+IG EXPP R)00,0pLG←IG+PP[2;FCF[I]]
V0←V0,T,' )',S1
→(1=Λ/' '=T+IG EXPP R)00,0pLG←IG+PP[2;FCF[I-1]]-PP[2;FCF[I]]
V0←V0,' ( ,T,' )',S2
→(1=Λ/' '=T+IG EXPP R)00,IG←IG+PP[2;FCF[I]]
→0,0pV←V0,T,0pSWE←1
IL22:→(1=Λ/' '=T+IG EXPP R)00
V0←'⊙(' ,T,' )',S2
→(1=Λ/' '=T+IG EXPP R)00,IG←IG+PP[2;FCF[I]]
→0,0pV←V0,T,0pSWE←1
LL11:→(1=Λ/' '=T+IG EXPP R)00
V0←(' ,S2,T,' )',S1
→(1=Λ/' '=T+IG EXPP R)00,0pLG←IG+PP[2;FCF[I]]
→0,0pV←V0,S2,T,0pSWE←1
IM:→(IMAX≥VL)00
IMAX←VL

V←ET1;VL;IL;S1;S2;V0;K;PR;TOPP
DET:→DET+VDT1DT,0pSWY←0
CO:→EXIT+VDO1DO
N2:→0,CF←0pV+CF
V1:→(EXIT+VDO1DO)×1LAG=LG+0,0pLI←0pVV+VV,CC,', '
V4:→EXIT+VDO1DO,0pLI←'-' ,VIG←LAG+I,0pVV←VV,CC,', '
PR:→EXP+VDO1DO,0pR←I1,I2,0pIG←LAG
R24:→EXP+VDO1DO,0pR←I1,I2,0pIG←LAG+I
EXIT:→0,0p□←'COMBINAISON DE FONCTIONS NON ADMISE PORTANT SUR : ',CC,V←' '
V:→IM,0pV←CC,'(I',LL,' )',0pVL←IG
C:→COEF
P:→IM,0pV←CC,'(I',LL,' )'-',CC,'(I-',(VVL),' )',0pVL←PR[2;FCF[I]]+IG,0pSWY←1
P:→IM,0pV←CC,'(I',LL,' )'/',CC,'(I-',(VVL),' )',0pVL←PR[2;FCF[I]]+IG,0pSWY←2
L:→IM,0pV←'LOG(' ,CC,'(I',LL,' )')',0pVL←IG,0pSWY←4
A:→IM,0pV←'APS(' ,CC,'(I',LL,' )')',0pVL←IG,0pSWY←4
PE:→IM,0pV←CC,'(I',LL,' )'/',CC,'(I-',(VPR[2;FCF[I]]+IG),' )'-',CC,'(I-',(VPR[2;FCF[I-1]]+IG),' )'/',CC,'(I-',(VVL+PR[2;FCF[I]]+PR[2;FCF[I-1]]+IG),' )'
PD:→IM,0pV←(' ,CC,'(I',LL,' )'-',CC,'(I-',(VPR[2;FCF[I]]+IG),' )')÷(' ,CC,'(I-',(VPR[2;FCF[I-1]]+IG),' )'-',CC,'(I-',(VVL+PR[2;FCF[I]]+PR[2;FCF[I-1]]+IG)'

```



```

LD:→IM,0oV+LOG('CC','(I',LL,')'-',CC,'(I-',(VVI+PP[2;FCF[I]]+LG),'))',0oSWY
+4
LE:→IM,0oV+LOG('CC','(I',LL,')/'',CC,'(I-',(VVI+PP[2;FCF[I]]+LG),'))',0oSWY
+4
LI:→IM,0oV+LOG('CC','(I',LL,')')-LOG('CC,'(I-',(VVI+PP[2;FCF[I-1]]+LG),'))
',0oSWY+1
LJ:→IM,0oV+LOG('CC','(I',LL,')')/LOG('CC,'(I-',(VVI+PP[2;FCF[I-1]]+LG),'))
',0oSWY+2
AVR:→APLPOL×1TST[5],0oK+I
OK:→0,0oV+('',PARAM1[STP1[DO]+1LGP1[DO]],(VPP[2;FCF[K]]),',',CC,')',0oSWE+0
AG:→AVR
AX:→AVR
AN:→AVR
EXP:→0,0oV+COMBINAISON DE FONCTIONS NON ADMISE PORTANT SUR : ',XX[(I1-1)+1
I2-I1-2],V+' '
E:→LS1,0oS1+('',0oS2+')'
CE:→EXP
DE:→LL11,0oS1+('-',S2+0oSWY+1
EE:→LL11,0oS1+('/',S2+0oSWY+2
IE:→LS1,0oS1+LOG('',0oS2+')',0oSWY+4
AE:→LS1,0oS1+APS('',0oS2+')',0oSWY+4
DEF:→LS12,0oS1+('-',0oS2+')',0oSWY+1
EDF:→LS12,0oS1+('/',0oS2+')',0oSWY+2
ID:→LI22,0oS1+LOG('',0oS2+')',0oSWY+4
LD:→LI22,0oS1+LOG('',0oS2+')',0oSWY+4
DIF:→LI11,0oS1+('-',0oS2+')LOG',0oSWY+1,0oI+I-1
EIF:→LI11,0oS1+('/',0oS2+')LOG',0oSWY+2,0oI+I-1
AVF:→AVR
AVF:→AVR
ACF:→AVR
MYF:→AVR
MNF:→AVR
→FYP
COFF:→FYIT×1~(A/CFE NIM)A0×oCF
→SWY+0,CF+0oV+VC,(V+'C('',CF,')'),',',
APLPOL:→OK,0oCC+PL1POL XX[I1+1I2-I1]
IS1:→(1=A/' '=T+LG EXPR1 B)00
→0,0oV+S1,T,S2,0oSWY+0
LS12:→(1=A/' '=T+LG EXPR1 B)00
V0+('(',T,')',S2
→(1=A/' '=T+LG EXPR1 B)00,0oLG+LG+PP[2;FCF[I]]
V0+V0,('(',T,')'),S1
→(1=A/' '=T+LG EXPR1 B)00,0oLG+LG+PP[2;FCF[I-1]]-PP[2;FCF[I]]
V0+V0,('(',T,')'),S2
→(1=A/' '=T+LG EXPR1 B)00,LG+LG+PP[2;FCF[I]]
→0,0oV+V0,('(',T,')'),0oSWY+1×(S1='-')+2×S1='/'
LI22:→(1=A/' '=T+LG EXPR1 B)00
V0+S1,('(',T,')'),S2
→(1=A/' '=T+LG EXPR1 B)00,LG+LG+PP[2;FCF[I]]
→0,0oV+V0,('(',T,')'),0oSWY+0
LI11:→(1=A/' '=T+LG EXPR1 B)00
V0+S2,('(',T,')'),S1
→(1=A/' '=T+LG EXPR1 B)00,0oLG+LG+PP[2;FCF[I]]
→0,0oV+V0,S2,('(',T,')'),0oSWY+1×(S1='-')+2×S1='/'
IM:→(IMAX>VL)00

```



LMAX+VI

0000

```

ERR←INIT;A
XX←(1-XXε' ')/XX
NP←\ (XY='(')-XX=')'
→(NP[0NP]≠0)0ODD
→(1=Λ/A+XXεLTR,LTRF,NUM,CAR)0ERR←0
NOT:→1-FPR+1,00□←'NOT ALLOWED CHARACTER ',(Λ)/XX
ODD:□←'NOT EVEN PARENTHESES ':',00FPR+1
→(0=Λ/A+XXεLTP,LTPF,NUM,CAR)0NOT

```

0000

```

INITIAL
FCT←'AVAGMYMN CDRLA '
STT← 0 2 4 6 8 10 11 12 13 14 15
IGT← 2 2 2 2 2 1 1 1 1 1 2
PAPAM1←'AVAGMYMN C[Λ÷0]'
STP1← 0 3 6 9 12 14 16 17 18 19
IGP1← 3 3 3 3 2 2 1 1 1 1
PAPAM2←'AVAGMYMN [Λ÷0]'
STP2← 0 4 8 12 16 18 19 20 21 22
IGP2← 4 4 4 4 2 1 1 1 1 1
NECT← 6 9 10
DECT← 7 8
MASK←1+MASK+2
VDT← 0 2 1 4 8 24
R C N VV( E E(
VDO← 0 6 7 8 9 10 708 807 907 908 709 809 1 1 2 3 4
R V C D E L A DE ED LD LE DL EL AV AGMYMN

```

0000

```

INITIAL1
FCT←'AVAGMYMN CDRLA '
STT← 0 2 4 6 8 10 11 12 13 14 15
IGT← 2 2 2 2 2 1 1 1 1 1 2
PAPAM1←'AV=ΛAV(AG=ΛAG(MY=ΛMY(MN=ΛMN( C(Λ÷LOG(ABS('
STP1← 0 7 14 21 28 31 33 34 35 39
IGP1← 7 7 7 7 3 2 1 1 4 4
PAPAM2←''
STP2←''
IGP2←''
NECT← 6 9 10
DECT← 7 8
MASK←1+MASK+2
VDT← 0 2 1 4 8 24
R C N VV( E E(
VDO← 0 6 7 8 9 10 708 807 907 908 709 809 1 1 2 3 4
R V C D E L A DE ED LD LE DL EL AV AGMYMN

```

0000

```

P←I ACCED
I←'I'
→I0×1 (XX[Y+1]='(') ∧ NP[Y+1]=NP[Y]+1
→P+0
EXIT:→P+2, 00[←'ERREUR IN IAG : ', XX[IND]
I0:→EXIT×1~YY[Y+Y+2]∈NUM
I1:→I2, 00I+L, XX[Y]
I2:→I1×1YY[Y+Y+1]∈NUM
→EXIT×1~YY[Y]∈')'
P+1, 002L

```

0000

```

ERR←LEFT;TST;0;I;A;CC;DEC;X;PO
PO←CC+00I+0
LPC←'DPL'==
TST←400
R TST←D,E,L,-
O←'DEC+0'
LOOP:→EXIT×1FORNES[2]≤I+I+1
→LCC×11=XX[I]∈LTP, NUM
→EXIT×10=v/A+IPC∈XX[I]
→(L1,L1,I1,I4,I5)[A+A/10A]
EXIT:→1-FPP+1, 00[←' IN ', XX[11+(XX∈')/10XX], 00[←'ERREUR : ', XX[1I]
ICC:→LOOP, 00CC+CC, YY[I]
I1:→EXIT×1TST[A]≠0
→I11×10≠0CC
CC←'1'
I11:→LOOP, 00TST[A]+1, 000+0, vA, CC+00PO+PO, CC, ', '
I4:→EXIT×1(TST[Y+A-2]=0)∨(TST[A]≠0)∨XX[I+I+1]≠'1'
→LOOP, 000+0, vA, 00TST[A]+1
I5:→EXIT×1((YY[1]∈NUM)∨v/CC∈'. '), 0020
→EXIT×11=LPT
→ST×1~((XX[I+1]='/' ) ∧ 'x'='-1+ZZ)∨(XX[I+1]='-') ∧ '+'='-1+ZZ
ZZ←-1+ZZ
ST:→FPP+0, 00VV←' ', CC, ', ', 00FORNES[1]+I

```

0000

```

ERR←LEFT1;TST;0;I;A;CC;DEC;X;PO
PO←CC+00I+0
LPC←'DEL'==
TST←400
R TST←D,E,L,-
O←'DEC+0'
LOOP:→EXIT×1FORNES[2]≤I+I+1
→LCC×11=XX[I]∈LTP, NUM
→EXIT×10=v/A+IPC∈XX[I]
→(L1,L1,I1,I4,I5)[A+A/10A]

```



```

EXIT:→1-EPR+1,0o[←' IN ' ,XX[11+(XXε=')/1oXX],0o[←'EPFEUR : ' ,XX[1I]
LCC:→LOOP,0oCC+CC,XX[I]
L1:→EXIT×1TST[A]≠0
→L11×10≠oCC
CC←'1'
L11:→LOOP,0oTST[A]+1,0o0+0,VA,CC+0oPO+PO,CC,', '
L4:→EXIT×1(TST[X+A-2]=0)∨(TST[A]≠0)∨XX[I+I+1]≠'1'
→LOOP,0o0+0,VA,0oTST[A]+1
L5:→EXIT×1((XX[1]εNUM)∨∨/CCε'. '),0o±0
→EXIT×11=LPT1
→ST×1~((XX[I+1]='/' )^'x'='1+ZZ)∨(XX[I+1]='-')^'+='1+ZZ
ZZ←1+ZZ
ST:→EPR+0,0oVV←' ,CC,',',0oBOPNES[1]←I

```

□□□□

```

V ER1←LPT:Y:Y1:Y2:Y3:A
* VAP,D,E,L,E-1,DL,DE,DE-1,LD,LE,LE-1,ED,EL,ED-1,EL-1
* 0,1,2,3, 24,13,12, 124,31,32, 324,21,23, 214, 234
X←CC,'[I]←',0oPO+±PO,'0,0'
Y1←CC,'[I-',A,']',0oA+±PO[1]
Y2←CC,'[I-',A,']',0oA+±PO[2]
Y3←CC,'[I-',A,']',0oA+±PO[1]+PO[2]
DEF:→DEF+ 0 1 3 2 24 13 12 124 31 32 324 21 23 214 234 1DEC
→(1,0oZZ+Y) oEP1+0
→(1,0oZZ+Y,Y1,'+') oEP1+0
→(1,0oZZ+Y,'*') oEP1+0
→(1,0oZZ+X,Y1,'x') oEP1+0
→(1,0oZZ+Y,Y1,'x1+') oEP1+0
→(1,0oZZ+X,'*(@',Y1,')+') oEP1+0
→(1,0oZZ+X,Y2,'x(' ,Y1,'÷',Y3,')+') oEP1+0
→(1,0oZZ+X,Y2,'x(' ,Y1,'÷',Y3,')+') oEP1+0
→(1,0oZZ+X,Y2,'++') oEP1+0
→(1,0oZZ+X,Y2,'x*') oEP1+0
→(1,0oZZ+X,Y2,'x1*') oEP1+0
→(1,0oZZ+X,Y2,'+(' ,Y1,'-',Y3,')x') oEP1+0
→(1,0oZZ+X,'*(@',Y1,')x') oEP1+0
→(1,0oZZ+X,Y2,'+(' ,Y1,'-',Y3,')x1+') oEP1+0
→(1,0oZZ+X,'*(@',Y1,')x1+') oEP1+0
ER1←1

```

□□□□

```

V ER1←LPT1:X:Y1:Y2:Y3:A
* VAP,D,E,L,E-1,DL,DE,DE-1,LD,LE,LE-1,ED,EL,ED-1,EL-1
* 0,1,2,3, 24,13,12, 124,31,32, 324,21,23, 214, 234
X←CC,'(I)=' ,0oPO+±PO,'0,0'
Y1←CC,'(I-',A,')',0oA+±PO[1]
Y2←CC,'(I-',A,')',0oA+±PO[2]
Y3←CC,'(I-',A,')',0oA+±PO[1]+PO[2]
DEF:→DEF+ 0 1 3 2 24 13 12 124 31 32 324 21 23 214 234 1DEC
→(1,0oZZ+X) oEP1+0
→(1,0oZZ+Y,Y1,'+') oEP1+0

```



```

→(1,0pZZ+X,'EXP(')oEP1+0,0oPAR+1
→(1,0pZZ+Y,Y1,'*')oEP1+0
→(1,0pZZ+Y,Y1,'*(1+')oEP1+0,0oPAR+1
→(1,0pZZ+Y,'EXP(LOG('Y1,')+')oEP1+0,0oPAR+1
→(1,0pZZ+Y,Y2,'*('Y1,','Y3,')+')oEP1+0,0oPAR+1
→(1,0pZZ+Y,Y2,'*('Y1,','Y3,')+')oEP1+0,0oPAR+1
→(1,0pZZ+Y,Y2,'+EXP(')oEP1+0,0oPAR+1
→(1,0pZZ+Y,Y2,'*EXP(')oEP1+0,0oPAR+1
→(1,0pZZ+Y,Y2,'*(1+EXP(')oEP1+0,0oPAR+2
→(1,0pZZ+Y,Y2,'+((('Y1,','Y3,')*')oEP1+0,0oPAR+1
→(1,0pZZ+Y,'EXP(LOG('Y1,')*')oEP1+0,0oPAR+1
→(1,0pZZ+Y,Y2,'+((('Y1,','Y3,')*(1+')oEP1+0,0oPAR+1
→(1,0pZZ+Y,'EXP(LOG('Y1,')*(1+')oEP1+0,0oPAR+2
EP1+1

```

```

000000

```

```

▽ MN←N MN X;Y
→ST×1(N<oY)∧N>0,0o2'Y←',vX
N←oY
ST:MN←L/((oY)-N)+Y

```

```

000000

```

```

▽ MX←N MX X;Y
→ST×1(N<oY)∧N>0,0o2'Y←',vX
N←oY
ST:MX←L/((oY)-N)+Y

```

```

000000

```

```

▽ FFS←PL1POL FY:I:A
→((I+0)=oA+(FYε'*')/1oFY)oS
R:→((oA)<I+I+1)oS
→P,0oFY+FY[1A[I]], '*' ,A[I]+FY
S:→((I+0)=oA+(FYε'*')/1oFY)oS2
C:→((oA)<I+I+1)oS2
→C,0oFY+FY[1A[I]-1], '*' ,A[I]+FY
S2:→((I+0)=oA+(FYε'*')/1oFY)oS3
D:→((oA)<I+I+1)oS3
→D,0oFY+FY[1A[I]-1], '/' ,A[I]+FY
] S3:FES←FY

```

```

000000

```

```

▽ TT←LAG TTE N;I;MAT;SA
I+0
TABLEAU
RANGE
DO:→((oMAT[2;])<I+I+1)oOD
R←MAT[2;I],MAT[3;I]-1

```

```

→SUITE×1B[1]×P[2]
→FIN,0o[←'ELEMENT VIDE : ',XX[1B[1]+1]
SUITE:→(TERM,FACT,EXPT)[N[1]+1]
TERM:→(1=Λ/' '=E+LAG TTF(1,P))oFIN,0oP+NP[B[1]]×Λ/1≤P+-1+NP[B[1]+1B[2]-B[1]
]
→DF×1~(SWEVSWY)×I<oMAT[1:]
F+(' ',F,')',0oSW+SWE+SWY+0
DE:ANYP F,0oY+FCUM
→DO,0oFCUM+Y
FACT:→(1=Λ/' '=E+LAG TTF(2,P))oFIN,0oP+NP[B[1]]×Λ/1≤P+-1+NP[B[1]+1B[2]-B[1]
]
→DF×1~(SWEVSWY)×I<oMAT[1:]
F+(' ',E,')',0oSW+SWE+SWY+0
DE:ANYP E,0oY+FCUM
→DO,0oFCUM+Y,0oSW+SWY+I≥2
EXPT:→(1=Λ/' '=Y+ANALYSE)oFIN+SWE+0
→D2×1SW+I≥2
→DO,0oSA+SWY,0oYCUM+Y
D2:→D3×1(SA=0)ΛI=2
→DO,0oY+YCUM+(' ',YCUM,')*',Y
D3:→DO,0oY+YCUM+YCUM,('*',Y
OD:→0,0oTT+Y
FIN:TT+F+F+ ' '
▽

```

000000

```

▽ TT+LAG TTF1 N:I:'AT:SA
SA+I+0
TABLEAU
DO:→((oMAT[2:])<I+I+1)oOD
F+MAT[2:I],MAT[3:I]-1
→SUITE×1B[1]×P[2]
→FIN,0o[←'ELEMENT VIDE : ',XX[1B[1]+1]
SUITE:→(TERM,FACT,EXPT)[N[1]+1]
TERM:→(1=Λ/' '=E+LAG TTF1(1,P))oFIN,0oP+NP[B[1]]×Λ/1≤P+-1+NP[B[1]+1B[2]-B[1]
]]
→DF×1~SWE×SWY×4
F+(' ',F,')'
DE:→DO,0oSWY+SWE+0,0oFCUM+Y+FCUM CONSTRUCT F
FACT:→(1=Λ/' '=E+LAG TTF1(2,P))oFIN,0oP+NP[B[1]]×Λ/1≤P+-1+NP[B[1]+1B[2]-B[1]
]]
→DF×1~(SWEVSWYε1)×SWY×4
F+(' ',E,')'
DE:→DO,0oSWY+SWE+0,0oFCUM+Y+FCUM CONSTRUCT E
EXPT:→(1=Λ/' '=Y+ANALYSE1)oFIN+SWE+0
→D2×1I>2
→DO,0oYCUM+Y,0oSA+SWY
D2:→D3×1~SAε1,2
YCUM+(' ',YCUM,')'
D3:→D4×1~SWYε1,2,0oSA+0
→DO,0oY+YCUM+YCUM,('*(' ',Y,')'
D4:→DO,0oY+YCUM+YCUM,('*',Y
OD:→0,0oTT+Y
FIN:TT+F+F+ ' '

```

v

000000

## TABLE OF CONTENTS

APIT	5	<u>ANALYSE</u>	9	<u>EXPP1</u>	15	<u>LPT</u>
DRIN	6	<u>ANALYSE1</u>	10	<u>ELTPE</u>	15	<u>LPT1</u>
LEGIN	7	<u>ANYP</u>	10	<u>ET</u>	16	<u>MN</u>
PL1T	7	<u>APLPOI</u>	11	<u>ET1</u>	16	<u>MY</u>
P11T	8	<u>APTEL</u>	13	<u>INIT</u>	16	<u>PL1POI</u>
TAPL	8	<u>AV</u>	13	<u>INITIAL</u>	16	<u>TPE</u>
TPAL	8	<u>POOL</u>	13	<u>INITIAL1</u>	17	<u>TPE1</u>
TPL1	9	<u>POOL1</u>	14	<u>LACGED</u>		
TRAD	9	<u>CONSTRUCT</u>	14	<u>LEFT</u>		
<u>AC</u>	9	<u>EXPP</u>	14	<u>LEFT1</u>		



BUMP



0 0 3 2 1 3 3 7 4

\*FM B16/1978/10

